PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-081118
(43)Date of publication of application : 28.03.1997

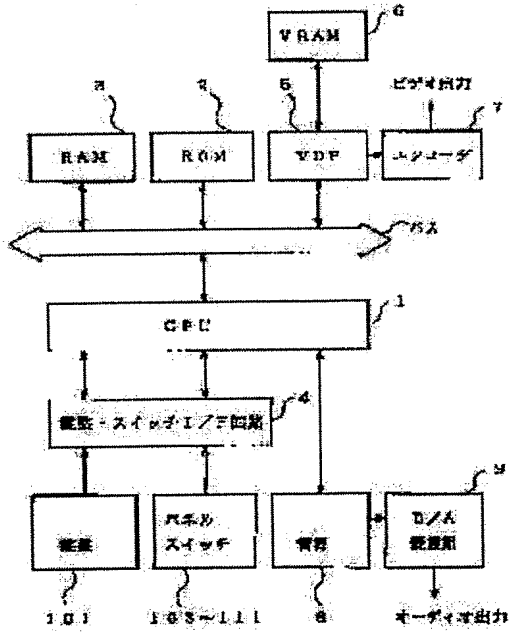
(51)Int.Cl.
G09G 5/36
G09G 5/36
A63F 9/22
G06F 3/14
G09G 5/00
G09G 5/00
G09G 5/06
G09G 5/14
G10H 1/00
H04N 7/18

(21)Application number : 07-259254
(22)Date of filing : 11.09.1995

(71)Applicant : CASIO COMPUT CO LTD
(72)Inventor : ISHIGURO SHIRO

(54) IMAGE CONTROL DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To improve the processing speed without using a high-speed CPU by making only the image a target of updating, of which a display mode is changed.
SOLUTION: In a character string alteration processing, 18 character strings, 6 lines of OBJB code, are read from a storage area of ROM 2 according to a cross key operation as the reference to a cursor position scrolled in accordance with value of a pointer register i, and when the read OBJB code is a space (blank), a transparent character (transparent color designation) is set, when it is not the case, the object B character (character image) specified by the OBJB code is set. Namely, with all the characters altered in display set to object B image, a corresponding OBJB code is designated in the case of altering display, and a transparent color may be designated in the case of erase display. Thus, a high speed image display is made possible.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision

of rejection]
[Date of extinction of right]

(19) 日本国特許庁 (JP)

(12) 公開特許公報（A）

(11) 特許出願公開番号

特開平９－８１１１８

(43) 公開日 平成 9年(1997)3 月 28日

(51)Int.Cl.⁶		識別記号	庁内整理番号	F I			技術表示箇所
Ｇ０９Ｇ	5/36	５２０	9377－５Ｈ	Ｇ０９Ｇ	5/36	５２０ Ｎ	
		５３０	9377－５Ｈ			５３０ Ｊ	
Ａ６３Ｆ	9/22			Ａ６３Ｆ	9/22	Ｂ	
Ｇ０６Ｆ	3/14	３１０		Ｇ０６Ｆ	3/14	３１０ Ａ	
Ｇ０９Ｇ	5/00	５１０	9377－５Ｈ	Ｇ０９Ｇ	5/00	５１０ Ｈ	

審査請求 未請求 請求項の数 6 ＦＤ （全 ３４ 頁） 最終頁に続く

(21) 出願番号 特願平7-259254

(22) 出願日 平成 7年(1995)9 月 11日

(71) 出願人 000001443
カシオ計算機株式会社
東京都新宿区西新宿2丁目6番1号

(72) 発明者 石黒 士郎
東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社羽村技術センター内

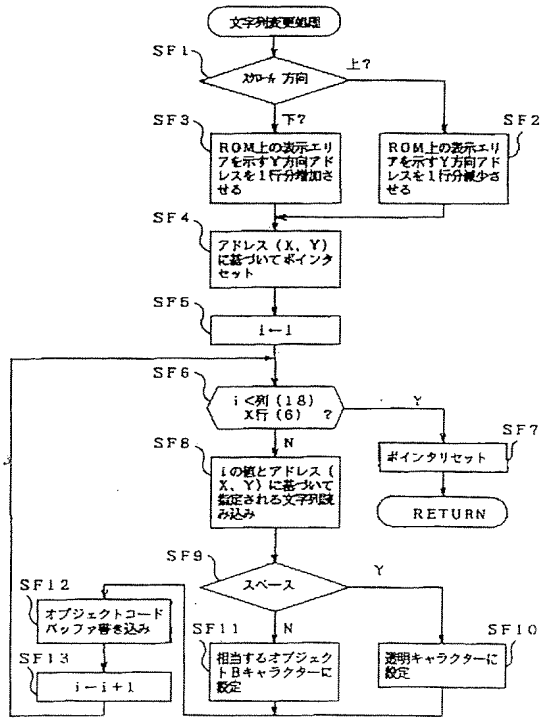
(74) 代理人 弁理士 鹿嶋 英實

(54)【発明の名称】画像制御装置

(57)【要約】

【課題】 コストアップを招致することなく高速な画面表示を実現する画像制御装置を実現する。

【解決手段】 画面タイトルなどの画面上で固定的に表示される文字をオブジェクトＡ画像で形成する一方、このオブジェクトＡ画像上で表示変更される文字を表示する表示領域（表示行）に、予め表示可能な最大文字数分のオブジェクトＢ画像を配置しておき、その文字を書き換える時にはその表示位置に対応するＯＢＪＢコード中のキャラクタネーム（文字種類）だけを変更し、文字を消去する際にはそのＯＢＪＢコード中のカラーブロックを透明指定させて非表示状態する。この結果、表示態様が変更される画像だけが更新対象となるから、高速なＣＰＵを用いずに処理速度の向上でき、結果的にコストアップを招致することなく処理速度の向上が可能になる。



【特許請求の範囲】

【請求項１】　表示態様を変更可能とした変更画像の表示領域を予め画面上で定めておき、この表示領域内に表示し得る数分の変更画像を配置する画像配置手段と、
前記表示領域内に配置される変更画像のいずれかを指定してその表示態様を変更する変更手段と、
この変更手段によって変更された表示態様に応じて変更画像を更新し、更新した変更画像を前記表示領域内での配置に従って画面表示を指示する表示制御手段とを具備することを特徴とする画像制御装置。

【請求項２】　画面上で固定的に表示される固定画像の種類、表示位置および表示色を指定する第１の属性データと、画面上で表示変更される変更画像の種類、表示位置および表示色を指定する第２の属性データとを記憶する属性データ記憶手段と、
前記第１および第２の属性データにそれぞれ対応付けられた画像データを記憶する画像データ記憶手段と、
前記第１および第２の属性データに応じて前記画像データ記憶手段から対応する画像データをそれぞれ読み出して前記固定画像上に前記変更画像を重ねて画面表示を指示する画像表示指示手段と、
前記変更画像を表示変更する指示を受けた場合、その指示に応じて前記第２の属性データを書き換える書き換え手段と、
この書き換え手段によって書き換えられた前記第２の属性データに対応する変更画像のみ表示変更を指示する表示制御手段とを具備することを特徴とする画像制御装置。

【請求項３】　前記書き換え手段は、前記第２の属性データに含まれる画像種類あるいは表示色のいずれか一方を書き換えることを特徴とする請求項２記載の画像制御装置。

【請求項４】　前記書き換え手段は、変更画像を空白にする場合、前記第２の属性データに含まれる表示色を透明に書き換えて非表示状態にすることを特徴とする請求項２記載の画像制御装置。

【請求項５】　前記第１の属性データは、対応する画像データをアドレス順に前記画像表示手段へ供給するか、あるいは逆順に供給するかを指定するフラグを備え、当該フラグの値に応じて当該画像表示手段がアドレス順に供給された通常の固定画像と、逆順に供給されて像が反転した反転画像とのいずれかを指定することを特徴とする請求項２記載の画像制御装置。

【請求項６】　前記固定画像は、複数の画像データから形成され、これら各画像データ毎に対応付けられる前記第１の属性データの表示色を変化させて表示態様を可変としたことを特徴とする請求項２記載の画像制御装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、例えば、テレビゲーム装置などに用いて好適な画像制御装置に関する。

【０００２】

【従来の技術】近年、テレビジョン受像機のＡＶ（オーディオ・ビジュアル）端子に接続され、遊技玩具や知育玩具などとして使用されるテレビゲーム装置が各種実用化されている。この種の装置は、一般にＣＰＵ，ＲＯＭ，ＲＡＭおよびＶＲＡＭ（ビデオＲＡＭ）等から構成される画像制御装置を備え、ＲＯＭに記憶されている静止画像および動画像の各画像データを、ＣＰＵの指示の下にＶＲＡＭに転送する一方、このＶＲＡＭに転送された各画像データを読み出し、これをビデオ信号に変換してテレビジョン受像機のＡＶ端子に供給することで、そのブラウン管上に背景像（静止画像）を表示しながら、ゲーム操作に応じて移動する複数のキャラクタを動画表示するようになっている。

【０００３】

【発明が解決しようとする課題】ところで、こうした画像制御装置では、基本的にパーソナルコンピュータ等と同様の画像制御を行う訳であるが、遊技玩具あるいは知育玩具として用いられる性格上、製品価格を出来るだけ低廉なものにする必要がある。この為、必然的に使用するハードウェアが制限されてしまい、例えば、画像メモリ（ＶＲＡＭ）容量を抑え、マウス等のポインティングデバイスを省略したものでは、所謂、ＧＵＩ（グラフィカル・ユーザー・インタフェース）環境と呼ばれる操作環境が実現し難いという現状にある。

【０００４】また、ＧＵＩ環境下で文字列を表示するには、ウインドウ（表示窓）中に表示される文字列をオブジェクトとして扱い、表示すべき各文字のキャラクタ番号とその表示座標とを個々に設定する。したがって、画面タイトルなどのように、画面上で固定的に表示される文字についても対応するキャラクタ番号と表示座標とを個々に管理しなければならず、さらに、表示中の文字を消すには対応するキャラクタ番号のオブジェクトを表示領域外へ配置するといった操作が必要になるため、これらにより画面を表示する際の処理が遅くなるという弊害が生じている。処理速度を向上させるには、高速なＣＰＵを用いれば良いが、上述したように、製品性格上、コストアップを招致することができず、これが技術的課題となっている。

【０００５】そこで、本発明は、上述した事情に鑑みてなされたもので、その主たる目的は、コストアップを招致することなく高速な画面表示を実現する画像制御装置を提供することにある。さらに、他の目的として、画像メモリ（ＶＲＡＭ）容量を抑え、マウス等のポインティングデバイスを省略した構成下でも疑似的なＧＵＩ環境を実現し得る画像制御装置を提供することにある。

【０００６】

【課題を解決するための手段】上記目的を達成するため、請求項１に記載の発明では、表示態様を変更可能と

した変更画像の表示領域を予め画面上で定めておき、この表示領域内に表示し得る数分の変更画像を配置する画像配置手段と、前記表示領域内に配置される変更画像のいずれかを指定してその表示態様を変更する変更手段と、この変更手段によって変更された表示態様に応じて変更画像を更新し、更新した変更画像を前記表示領域内での配置に従って画面表示を指示する表示制御手段とを具備することを特徴としている。

【０００７】また、請求項２に記載の発明では、画面上で固定的に表示される固定画像の種類、表示位置および表示色を指定する第１の属性データと、画面上で表示変更される変更画像の種類、表示位置および表示色を指定する第２の属性データとを記憶する属性データ記憶手段と、前記第１および第２の属性データにそれぞれ対応付けられた画像データを記憶する画像データ記憶手段と、前記第１および第２の属性データに応じて前記画像データ記憶手段から対応する画像データをそれぞれ読み出して前記固定画像上に前記変更画像を重ねて画面表示を指示する画像表示指示手段と、前記変更画像を表示変更する指示を受けた場合、その指示に応じて前記第２の属性データを書き換える書き換え手段と、この書き換え手段によって書き換えられた前記第２の属性データに対応する変更画像のみ表示変更を指示する表示制御手段とを具備することを特徴としている。

【０００８】上記請求項２に従属する好ましい実施態様として、請求項３に記載の発明によれば、前記書き換え手段は、前記第２の属性データに含まれる画像種類あるいは表示色のいずれか一方を書き換えることを特徴とする。

【０００９】また、請求項４に記載の発明では、前記書き換え手段は、変更画像を空白にする場合、前記第２の属性データに含まれる表示色を透明に書き換えて非表示状態にすることを特徴とする。

【００１０】請求項５に記載の発明では、前記第１の属性データは、対応する画像データをアドレス順に前記画像表示手段へ供給するか、あるいは逆順に供給するかを指定するフラグを備え、当該フラグの値に応じて当該画像表示手段がアドレス順に供給された通常の固定画像と、逆順に供給されて像が反転した反転画像とのいずれかを指定することを特徴とする。

【００１１】さらに、請求項６に記載の発明では、前記固定画像は、複数の画像データから形成され、これら各画像データ毎に対応付けられる前記第１の属性データの表示色を変化させて表示態様を可変としたことを特徴としている。

【００１２】本発明では、表示態様を変更可能とした変更画像の表示領域を予め画面上で定め、画像配置手段がこの表示領域内に表示し得る数分の変更画像を配置しておくと、変更手段によって表示領域内に配置される変更画像のいずれかを指定してその表示態様を変更すると、

変更された表示態様に応じて変更画像が更新され、これが表示制御手段によって前記表示領域内での配置に従って画面表示される。つまり、表示態様が変更される画像だけが更新対象となるから、高速なＣＰＵを用いることなく処理速度の向上を図ることが可能となり、換言すれば、コストアップを招致することなく処理速度の向上が可能になる。

【００１３】

【発明の実施の形態】本発明による画像制御装置は、例えば、テレビジョン受像機と接続し、そのブラウン管上に演奏形態を表示し、その中から選択した形態で演奏情報を自動演奏させたり、あるいは通常の楽器のように鍵盤操作に応じた楽音を発生する楽音制御装置などに適用され得る。以下では、こうした楽音制御装置を実施例とし、図面を参照して説明する。

【００１４】Ａ．実施例の外観

先ず、図１を参照して実施例の外観について説明する。図１は、本発明による画像制御装置が適用された楽音制御装置１００の外観を示す。この図において１０１は各鍵毎に鍵スイッチが配設された鍵盤であり、演奏者による押離鍵操作に応じたオンオフ信号を発生する。１０２～１１１はそれぞれパネル面に配設されるスイッチであり、１０２はこの装置１００の電源をオンオフするための電源スイッチ、１０３は発生楽音の音量を調整する音量スイッチである。１０４は自動演奏時あるいは自動伴奏時の演奏テンポを調整するテンポスイッチである。１０５は各演奏パート毎に所定の音色をアサインする音色スイッチ、１０６は自動演奏時あるいは自動伴奏時におけるリズムパートのリズム種類をセットするリズムスイッチである。

【００１５】１０７は自動演奏あるいは自動伴奏を停止させる際に操作されるストップスイッチである。１０８は自動演奏データの種類を選択する際に操作されるソングスイッチである。１０９は上下左右キーから構成される十字キーであり、後述するディスプレイターミナルＤＰ側に画面表示されるカーソルを上下左右へ移動させる際に操作される。１１０はエスケープキーであり、設定取消し時やカーソル位置を戻す際に操作される。１１１はエンターキーであり、確定入力する際に操作される。ＳＰは装置本体のパネル前面に配設される内蔵スピーカである。ＯＵＴ１，ＯＵＴ２はそれぞれパネル背面側に設けられたオーディオ出力端子、ビデオ出力端子である。これら出力端子ＯＵＴ１，ＯＵＴ２から出力されるオーディオ出力およびビデオ出力は、テレビジョン受像機あるいは周知のディスプレイターミナルＤＰに接続され、映像および音声（楽音）が再生される。

【００１６】Ｂ．実施例の構成

（１）全体構成

次に、図２を参照して実施例の全体構成について述べる。なお、この図において、図１の外観で示した部分と

共通する要素には同一の番号を付している。図２において、１はＣＰＵであり、スイッチ走査に基づきパネルスイッチ操作や押離鍵操作に対応するイベントを検出し、検出したイベント内容に応じて後述する構成要素５，６，７からなる表示制御部と、構成要素８，９からなる楽音制御部とを制御する。

【００１７】２はこのＣＰＵ１において実行される各種制御プログラムの他、画像表示するためのオブジェクト画像データ、オブジェクト画像データの表示座標位置や種類などの属性を表わすオブジェクトコードデータ、画面背景を形成するバックグラウンド画像データ、これら画像データを色データ（ＲＧＢデータ）に変換するカラールックアップテーブル、あるいは自動演奏用のソングデータなどが記憶されているＲＯＭである。このＲＯＭ２に格納される主要なデータについては追って詳述する。

【００１８】３はＲＡＭであり、上記ＣＰＵ１のワークエリアとして各種レジスタエリアが設けられ、演算結果やフラグ値などが一時記憶される。４は鍵盤およびスイッチインタフェース回路である。このインタフェース回路４は、パネル面（図１参照）に配設される各種操作スイッチ１０３～１１１のオンオフ操作や、あるいは鍵盤１０１の各鍵毎に設けられている鍵スイッチが押離鍵操作によりオンオフ操作される際のスイッチイベントを発生してＣＰＵ１に供給する。

【００１９】５は各種論理演算素子から構成されるビデオ・ディスプレイ・プロセッサ（以下、ＶＤＰと記す）であり、周知のＣＲＴコントローラとして機能する。ＶＤＰ５は、ＣＰＵ１の指示の下に表示制御する機能を担うものであって、ＲＯＭ２に格納される画像データを自身内部のキャラジェネメモリ（後述する）あるいは後述のＶＲＡＭ６へＤＭＡ転送する一方、これら各メモリに格納された画像データの内から表示すべき画像データを抽出してその表示形態や表示位置を定める表示制御処理を行い、この処理が施された画像データを表示色を表わすＲＧＢデータに変換して出力する。なお、このＶＤＰ５の構成については追って述べる。

【００２０】７はエンコーダであり、ＶＤＰ５より出力されるＲＧＢデータに垂直／水平同期信号を重畳してコンポジットビデオ信号を発生する。このコンポジットビデオ信号は、テレビジョン受像機あるいはディスプレイターミナルＤＰの映像入力端子に供給されることによって、ＶＤＰ５により表示制御された画像がＣＲＴ上に表示される。８は周知の波形メモリ読み出し方式で構成される音源であり、押離鍵操作に応じてＣＰＵ１が発生するキーオン／キーオフ、あるいはベロシティ等の演奏情報、あるいはＣＰＵ１が前述したＲＯＭ２から読み出した自動演奏情報（ソングデータ）に基づき、波形メモリから対応する波形データを読み出して楽音データを発生する。９は音源８から出力される楽音データをアナログ

形式のオーディオ出力信号に変換して出力するＤ／Ａ変換器である。このＤ／Ａ変換器９から出力されるオーディオ出力信号は内蔵スピーカＳＰより放音されるか、前述した出力端子ＯＵＴ１を介してディスプレイターミナルＤＰ側のスピーカから楽音として放音される。

【００２１】（２）ＲＯＭ２のデータ構成

次に、図３～図１６を参照してＲＯＭ２に格納される主要データの構成について説明して行く。ＲＯＭ２には、各種画面を形成するオブジェクトの属性、すなわち、オブジェクト種類およびその表示座標位置を表わすオブジェクトコードデータが記憶されるオブジェクト記憶エリアE_{OBJ}、これらオブジェクトコードに対応したオブジェクト画像データや画面背景を形成するバックグラウンド画像データが記憶される画像データ記憶エリアE_{IMAGE}および各画像データを色データ（ＲＧＢデータ）に変換するカラールックアップテーブルが記憶されるルックアップテーブル記憶エリアE_{CLT}が設けられている。以下、これら記憶エリアのデータ構成について述べる。

【００２２】①オブジェクト記憶エリアE_{OBJ}の構成

図３～図４を参照してオブジェクト記憶エリアE_{OBJ}の構成について説明する。この記憶エリアE_{OBJ}では、後述する各種画面毎のオブジェクトコードが割り付けて記憶されている。オブジェクトコードは、画面上で固定的に表示されるオブジェクトＡ、あるいは前述の十字キー１０９やエンターキー１１１の操作に応じて画面上で移動表示されるオブジェクトＢの２種類からなる。すなわち、初期画面用オブジェクトコードＯＢＪ１と、トータルコントロール画面用オブジェクトコードＯＢＪ５とには、それぞれ画面上で固定的に表示されるオブジェクトＡ用コードＯＢＪ1_{OBJA}，ＯＢＪ5_{OBJA}のみが記憶される。他の画面に対応するオブジェクトコードＯＢＪ２，ＯＢＪ３，ＯＢＪ４，ＯＢＪ６およびＯＢＪ７では、画面上で固定的に表示されるオブジェクトＡと、移動表示されるオブジェクトＢとの２種類のコードが記憶されている。

【００２３】ここで、図５および図６を参照し、上記形態で格納されるオブジェクトＡコードＯＢＪＡおよびオブジェクトＢ用コードＯＢＪＢのデータ形式について説明しておく。ＯＢＪＡ／ＯＢＪＢコードは、１ワードが１６ビット長のワードＷ０～Ｗ２の３ワードで１つのオブジェクト属性を形成している。オブジェクト属性とは、対応する画像データの大きさ（領域）、表示位置座標、キャラクタネームおよび使用するカラーブロック（後述する）などを表わす。

【００２４】図５あるいは図６に図示するように、ＯＢＪＡコード／ＯＢＪＢコードのワードＷ０，Ｗ１の下位９ビット（ビット０～ビット８）がオブジェクト面上における表示位置座標（Ｘ、Ｙ）を表わす。オブジェクト面とは、（０，０）～（５１１，５１１）の座標領域で

定義されるドット平面である。オブジェクト面ＯＢＪＡ（あるいはＯＢＪＢ）において、上記原点（０，０）を共有する横３３６ドット、縦２２４ドット（走査ライン）の領域が表示面となる。なお、表示位置座標（Ｘ、Ｙ）は、オブジェクト領域における左上隅の位置を表わしている。この表示位置座標（Ｘ、Ｙ）で定義されるオブジェクト位置が、上述の表示面の領域に入っていない場合には、そのオブジェクトは非表示となる。

【００２５】ＯＢＪＡコードにおけるワードＷ１のビット９には、対応する画像データを反転するか否かを表わす反転フラグＸがセットされる。このフラグＸは、オブジェクト画像データをラインバッファへ格納する際の書込み順序をアドレス順するか、あるいはそれと逆順にするかを指定するものであり、「０」の時にアドレス順を指定し、「１」の時に逆順を指定する。したがって、アドレス順に書込んだ通常の画像と、逆順に書込んだ画像とは像の左右が逆となり、両画像は鏡像関係になる。つまり、１つの画像データを通常に表示するものと、反転表示するものとの２画像を形成でき、このようにすることで、メモリ容量の少ないシステムにおいて表示し得る画像の種類を増やすことが可能になっている。

【００２６】オブジェクト領域の大きさは、最少８ドット×８ドットで形成され、その領域はワードＷ０，Ｗ１の上位４ビット（ビット１２～ビット１５）で縦横のサイズ（Ｘサイズ、Ｙサイズ）が可変指定され得る。例えば、Ｘサイズが「０」の時はオブジェクトの横幅が８ドット、「１」の時は１６ドット幅、「１５」の時に最大の１２８ドット幅となる。ワードＷ２では、その上位４ビット（ビット１２～ビット１５）が後述するカラールックアップテーブル部６５のカラーブロックを指定して表示色の種類を決め、ビット１２以降の下位１２ビットにはオブジェクトのキャラクタ種類を表わすキャラクタネームがセットされる。なお、カラーブロックとは、後述するカラールックアップテーブルＣＬＴを所定色数毎にブロック化してカラーコードをアサインしたものを言う。

【００２７】②画像データ記憶エリアE_{IMAGE}の構成

画像データ記憶エリアE_{IMAGE}では、上述した記憶エリアE_{OBJ}と同様、図７に示すように、各種画面毎の画像データが割り付けて記憶されている。画像データは、画面背景を形成するバックグラウンド画像データと、上記オブジェクトＡコードにより種類および表示座標位置が指定されるオブジェクトＡ画像データと、オブジェクトＢコードにより種類および表示座標位置が指定されるオブジェクトＢ画像データとの３種類に区分される。すなわち、図７に図示する通り、各画面の画像データＩＭ１～ＩＭ５は、上記３種類の画像データのいずれかが割り当てられて画面を形成する。３種類の画像データは、それぞれＢＧ画面、ＯＢＪＡ画面およびＯＢＪＢ画面を形成し、これら画面が重なり合って１つの表示画面を形成する。本実施例の場合、ＯＢＪＢ画面に最も高い優先順位が与えられ、続いて、ＯＢＪＡ画面、ＢＧ画面の順になる。したがって、重なる順序としては最も手前にＯＢＪＢ画面が配置され、その後にＯＢＪＡ画面、ＢＧ画面の順に奥側へ配置される。

【００２８】オブジェクトＢ画像データは、図８に図示するように、縦方向８ドット、横方向８ドットからなるセル単位で形成される。１セルを構成する各ドットには、１ビットのカラーコードが割り当てられ、それが「０」の場合には透明コードとなる。１セル分のカラーコードは、図９に示す通り、１ワードが１６ビット長のワードＣ０～Ｃ３の都合４ワードで表現される。すなわち、１セル内のドット行ｄ０～ｄ７のカラーコードがワードＣ０の上位８ビットに格納され、次のドット行ｄ１０～ｄ１７がワードＣ０の下位８ビットに格納される。以後、各ドット行ｄ２０～ｄ２７，ｄ３０～ｄ３７，…，ｄ６０～ｄ６７，ｄ７０～ｄ７７が同様にワードＣ１～Ｃ３に逐次格納される。

【００２９】一方、オブジェクトＡ画像データは、図１０に図示するように、上記オブジェクトＢ画像データと同様に縦方向８ドット×横方向８ドットを１セル単位として形成されているが、この場合、各ドットには４ビットのカラーコードが割り当てられ、透明色を含む１６色表示される画像となる。オブジェクトＯＢＪＡを形成する１セル分のカラーコードは、図１１に示す通り、１ワードが１６ビット長のワードＤ０～Ｄ１５の都合１６ワードで表現される。すなわち、１セル内のドットｄ００～ｄ０３のカラーコードがワードＤ０に格納され、次のドットｄ０４～ｄ０７がワードＤ１に格納される。以後、各ドットｄ１０～ｄ１３，ｄ１４～ｄ１７，…，ｄ７４～ｄ７７が同様にしてワードＤ１～Ｄ１５へ逐次格納される。

【００３０】③ルックアップテーブル記憶エリアE_{CLT}の構成

ルックアップテーブル記憶エリアE_{CLT}では、図１２に図示する通り、各種画面毎の画像データに対応するカラールックアップテーブルが記憶される。すなわち、初期画面用ＣＬＴデータＣＬＴ１は上述した初期画面用画像データ$IM1_{BG}$，$IM1_{OBJA}$をそれぞれ色データに変換するＢＧ画面用ルックアップテーブル$CLT1_{BG}$と、ＯＢＪＡ画面用ルックアップテーブル$CLT1_{OBJA}$とに分れており、こうしたテーブル形態は他の画面用ＣＬＴデータＣＴ２～ＣＬＴ５においても同様である。

【００３１】ここで、図１３～図１６を参照してカラールックアップテーブルの代表例について説明する。まず、図１３はバンド画面用ＣＬＴデータＣＬＴ２におけるＯＢＪＡ画面用ルックアップテーブル$CLT2_{OBJA}$のテーブル構成を示す図である。このテーブル$CLT2_{OBJA}$は、カラーブロック番号＃０～＃１５で指定される１６種類のルックアップテーブルを備え、各テーブルに

は１５色分の色データＤ１～Ｄ１５が格納されている。したがって、このテーブルの読み出しアドレスとなる画像データは、透明色を含む最大１６色でカラー表示され得る。

【００３２】このようなテーブルは、バンド画面中のＯＢＪＡ画面を形成する各オブジェクトＯＢＪＡ毎に割り当てられる。すなわち、前述したＯＢＪＡコードのワードＷ２における上位４ビットに格納される「カラーブロック値」（図５参照）により指定される。本テーブル$CLT2_{OBJA}$の場合、カラーブロック番号＃０～＃１１にバンドメンバーのポートレイト像（後述する）を表示するテーブルが割り当てられ、残りのカラーブロック番号＃１２～＃１５にはタイトル表示色や文字色などが割り当てられている。

【００３３】次に、図１４はジャンルセレクト画面用ＣＬＴデータＣＬＴ３におけるＯＢＪＡ画面用ルックアップテーブル$CLT3_{OBJA}$のテーブル構成を示す図である。この図に示すように、後述するジャンルセレクト画面において固定的に表示されるオブジェクトＯＢＪＡ、すなわち、ウインドウ（表示窓）、ウインドウタイトルおよびスクロール表示色などがそれぞれ個別のカラーブロックとして割り当てられている。これに対し、ジャンルセレクト画面において移動表示されるオブジェクトＯＢＪＢには、図１５に示すテーブル$CLT3_{OBJB}$がアサインされる。この場合、オブジェクトＯＢＪＢは、１ビットのカラーコードが割り当てられる「文字」なので、カラーブロック番号＃１に「黒」が割り当てられ、カラーブロック番号＃２に「ピンク」が割り当てられている。なお、カラーブロック番号＃０が割り当てられた場合には透明となる。

【００３４】図１６は、トータルコントロール画面用ＣＬＴデータＣＬＴ４におけるＯＢＪＡ画面用ルックアップテーブル$CLT4_{OBJA}$のテーブル構成を示す図である。この図に示すように、後述するトータルコントロール画面において固定的に表示される「ボタン」を形成するオブジェクトＯＢＪＡを、”ノーマル”、”セレクト”および”アクティブ”の３状態表示するようカラーブロックをアサインしている。なお、”ノーマル”、”セレクト”および”アクティブ”の３状態表示については追って説明する。

【００３５】（３）ＶＤＰ５の構成

次に、図１７を参照してＶＤＰ５の構成について説明する。この図において、５０はＣＰＵバスに接続されてＣＰＵ１とデータを授受するＣＰＵインタフェース回路である。５１はＶＲＡＭ６とのデータ授受を管理するＶＲＡＭインタフェースである。ＶＲＡＭ６には、ＣＰＵ１の指示の下に、これらインタフェース回路５０，５１をそれぞれ介してバックグラウンド画像データおよびオブジェクトＡ画像データが書込まれる。また、ＶＲＡＭ６からこれら画像データを読み出す場合には、書込み時と同様、上記インタフェース回路５０，５１を介してＣＰＵ１側から読み出しアドレスが与えられ、これに応じて読み出される画像データは後述のコントローラ５９，６３に供給される。

【００３６】５２は、画面上に表示されるオブジェクト数分のＯＢＪＡコード、ＯＢＪＢコードが格納されるオブジェクトメモリである。ＯＢＪＡコード、ＯＢＪＢコードとは、図５および図６に図示したように、オブジェクトＯＢＪＡを形成するオブジェクトＡ画像データと、オブジェクトＯＢＪＢを形成するオブジェクトＢ画像データとの各属性を表わすデータである。５３はＶＤＰ５の動作モードを決めるコントロールレジスタである。コントロールレジスタ５３は、全１６ビット長のレジスタであり、ＣＰＵ１により各ビット位置にセットされるレジスタ値に応じて表示制御形態が設定される。

【００３７】５４はキャラジェネメモリであり、オブジェクトＯＢＪＢを形成するオブジェクトＢ画像データが記憶される。このメモリ５４に格納される画像データは、「文字」を形成するものであって、図８に示したように、縦方向８ドット、横方向８ドットからなるセル単位で形成される。１セルを構成する各ドットには、１ビットのカラーコードが割り当てられ、それが「０」の場合には透明コードとなる。一方、前述したＶＲＡＭ６に格納されるオブジェクトＯＢＪＡを形成するオブジェクトＡ画像データは、図１０に示したように、上記オブジェクトＢ画像データと同様に縦方向８ドット×横方向８ドットを１セル単位として形成されているが、この場合、各ドットには４ビットのカラーコードが割り当てられ、透明色を含む１６色表示可能な画像となる。

【００３８】５５はＣＰＵ１の指示に応じてＤＭＡ転送制御するＤＭＡコントローラである。ＤＭＡコントローラ５５は、ＣＰＵ１からの転送指示に応じて前述したＲＯＭ２から転送対象データ、つまり、バックグラウンド画像データ、オブジェクトＡ，Ｂ画像データおよびＯＢＪＡ／ＯＢＪＢコードのいずれかを、転送指定先となるＶＲＡＭ６，オブジェクトメモリ５２あるいはキャラジェネメモリ５４のいずれかにブロック転送する。５６はオブジェクトリードコントローラであり、ディスプレイ側の水平走査ライン位置に対応するオブジェクト属性を前述したオブジェクトメモリ５２から読み出し、読み出したオブジェクト属性を、コントロールレジスタ５３が指定するオブジェクト種類の割り当て形態に応じた順序にソートし、その結果を後述するスタックＢメモリ５８、スタックＡメモリ５７にストアする。

【００３９】すなわち、このコントローラ５６では、オブジェクトメモリ５２に格納されているＯＢＪＡコード／ＯＢＪＢコードの内から、水平走査ラインが更新される度に、その走査ライン位置に合致するＹ座標およびＹサイズのＯＢＪＡ／ＯＢＪＢコードを抽出する。そして、抽出したＯＢＪＡ／ＯＢＪＢコードのＸ座標および

Ｘサイズと、コントロールレジスタ５３が指定する表示優先順位とに基づいて１水平走査ライン上の表示順序を求める。次に、求めた順序でＯＢＪＡコードをスタックＡメモリ５７に、ＯＢＪＢコードをスタックＢメモリ５８にそれぞれ格納する。

【００４０】したがって、スタックＡメモリ５７およびスタックＢメモリ５８では、それぞれ１水平走査期間において表示され得るオブジェクト属性、つまり、ＯＢＪＡ／ＯＢＪＢコードが記憶されることになり、これらメモリ５７，５８に記憶されるＯＢＪＡコード、ＯＢＪＢコードは、後述するコントローラ５９，６０によって表示順に読み出される。ラインバッファＡコントローラ５９は、上記スタックＡメモリ５７において表示順に記憶されているＯＢＪＡコードを順番に読み出し、当該ＯＢＪＡコード中に含まれる「キャラクタネーム」に対応するオブジェクトＡ画像データをＶＲＡＭ６から読み出す。

【００４１】また、このコントローラ５９は、ＯＢＪＡコード中に含まれるＸ座標値を書込みアドレスとして、ＶＲＡＭ６から読み出したオブジェクトＡ画像データをラインバッファＡ６１（後述する）に書込む。オブジェクトＡ画像データをラインバッファＡに書込む際、ＯＢＪＡコードにおけるワードＷ１のビット９に位置する反転フラグＸが「１」の場合には、書込みアドレスを逆順とし、一方、反転フラグＸが「０」の時には、アドレス順に書込む。これにより、像の左右が反転する鏡像関係の２画像を発生し得るようになっている。また、ラインバッファＡ６１に書込まれるオブジェクトＡ画像データには、ＯＢＪＡコードから抽出したカラーブロック値（図３参照）が付与される。

【００４２】ラインバッファＢコントローラ６０は、スタックＢメモリ５８において表示順に記憶されているＯＢＪＢコードを順番に読み出し、当該ＯＢＪＢコード中に含まれる「キャラクタネーム」に対応するオブジェクトＢ画像データを、キャラジェネメモリ５４から読み出してラインバッファＢ６２（後述する）に書込む。この書込みの際には、ＯＢＪＢコード中のＸ座標値が書込みアドレスとして用いられる。また、ラインバッファＢ６２に書込まれるオブジェクトＢ画像データには、ＯＢＪＢコードから抽出した「カラーブロック」（図４参照）が付与される。

【００４３】ラインバッファＡ６１、ラインバッファＢ６２には、それぞれ１水平走査ライン分の画像データを一時記憶するラインバッファメモリが複数ライン分設けられており、少なくとも、現在の水平走査に同期して表示される走査ライン分の画像データの他、次の水平走査に同期して表示される走査ライン分の画像データをも一時記憶するように構成されている。また、ラインバッファＡ６１、ラインバッファＢ６２では、水平走査に同期してバッファメモリを交互に切換えてライン読み出し、あるいはライン書込みするように上述したコントローラ５９，６０により制御される。つまり、一方のバッファメモリからキャラクタデータを読み出しながら、他方のバッファメモリにキャラクタデータを書込むことで、見掛け上の処理速度を向上している。

【００４４】６３はＢＧコードリードコントローラであり、ＶＲＡＭ６に格納されているバックグラウンド画像データを読み出して後段のＢＧデータレジスタ６４に格納する。このレジスタ６４に格納されるバックグラウンド画像データは、画面背景を形成する静止画像となる。６５は画面優先順位を付与する一方、優先順位が付与された各画面の画像データを色データ（ＲＧＢデータ）に変換して出力するカラールックアップテーブル部である。このテーブル部６５では、先ず、上述したラインバッファＡ６１、ラインバッファＢ６２およびＢＧデータレジスタ６４から水平走査に同期して出力されて来る各画像データ（オブジェクトＡ，Ｂ画像データおよびバックグラウンド画像データ）に優先順位を付与する。画面の重なり順序を表わす優先順位は、前述したコントロールレジスタ５３のレジスタ値に応じて定まる。

【００４５】テーブル部６５では、オブジェクトＡ画像に対応するカラールックアップテーブルデータを記憶する記憶エリアＣＬＴＡと、オブジェクトＢ画像に対応するカラールックアップテーブルデータを記憶する記憶エリアＣＬＴＢとに分れており、これら各記憶エリアには、各画像データに対応するオブジェクトコードに含まれる「カラーブロック値」で指定される種類のカラールックアップテーブルデータがＤＭＡ転送されて来る。そして、このテーブル部６５は、画面優先順位が付与された各画像データに対応するオブジェクトコード中の「カラーブロック値」に従って対応するカラーブロック番号のカラールックアップテーブルを選択し、選択したテーブルに基づき各画像データを色データ（ＲＧＢデータ）に変換して出力する。

【００４６】６６はＤ／Ａ変換器であり、上記カラールックアップテーブル部６５から出力される色データ（ＲＧＢデータ）を各色ＲＧＢ毎の色信号に変換して出力する。なお、各色信号は前述したエンコーダ７において水平／垂直同期信号が重畳されてコンポジットビデオ信号となる。６７は水晶発振子Ｘ’ｔａｌの原振クロックを逓倍発振して水平同期／垂直同期クロックを発生し、これをＶＤＰ５内の各部に供給する同期信号発生部である。

【００４７】（４）画面構成

次に、ＲＯＭ２に格納される各種データに基づいてＶＤＰ５が形成する各種画面の構成について説明する。前述したように、ＲＯＭ２には、「初期画面」、「バンド画面」、「バンドメンバー変更画面」、「ジャンルセレクト画面」、「トータルコントロール画面」および「バンドセーブ画面」を形成するためのオブジェクトコード、

画像データおよびルックアップテーブルデータが格納されている訳であるが、ここではこれらの内、代表的なものとして「バンド画面」、「バンドメンバー変更画面」、「ジャンルセレクト画面」および「トータルコントロール画面」の構成について述べる。

【００４８】①バンド画面の構成

バンド画面は、ディスプレイ上に表示される表示領域Ａと、この表示領域Ａの下側に位置し、ディスプレイに表示されずに不可視となる表示領域Ｂとから構成されている。なお、図１８は表示領域Ａの一態様を図示したものである。この表示領域Ａにおいては、タイトルバーやタイトル文字、選択ボタンあるいはミュージシャンのポートレイト像など表示変化しない部分がオブジェクトＡ画像（ＯＢＪＡ）で形成され、一方、このオブジェクトＡ画像上で、十字キー１０９やエンターキー１１１の操作に応じて表示変化する文字列ＣＨＲ１～ＣＨＲ８等がオブジェクトＢ画像（ＯＢＪＢ）で形成されている。

【００４９】表示領域Ｂでは、表示領域Ａにおいて表示使用されていない選択ボタン、タイトルバーおよびポートレイト像がオブジェクトＡ画像として退避されており、十字キー１０９の選択操作に応じて表示選択された時に、対応するＯＢＪＡコード中の表示座標位置を書き換えて表示領域Ａ側に移して表示するようになっている。また、表示領域Ａ側に置かれる各オブジェクトＡ画像は、前述したカラールックアップテーブル$ＣＬＴ２_{OBJA}$（図１３参照）の各カラーブロック＃０～＃１５に対応付けられ、指定カラーブロック中の色データで表示される。

【００５０】②バンドメンバー変更画面の構成

次に、図１９は、上述のバンド画面においてメンバー変更するように操作した際に表示されるバンドメンバー変更画面を示す図である。バンドメンバー変更画面では、バンド画面の下半分に表示される選択項目表示エリアを、表示領域Ｂに移動させる一方、その替わりに表示領域Ｂに配置されていた複数人数分のポートレート像の内から３人分のポートレート像を選択して表示する。図１９は、そうした状態を図示したものである。ところで、図１９に示すポートレート像Ｐ１、Ｐ２は同一のオブジェクトＡ画像データによって形成されるものであり、その一方の像を発生する際に前述したＯＢＪＡコード中の反転フラグＸを「１」にセットすることで像の左右が反転する鏡像関係の２画像を得ている。なお、このようなバンドメンバー変更画面においても、表示変化しない部分がオブジェクトＡ画像（ＯＢＪＡ）で形成され、表示変化する文字列等がオブジェクトＢ画像（ＯＢＪＢ）で形成されている。

【００５１】③ジャンルセレクト画面の構成

ジャンルセレクト画面では、図２０に示すように、画面中央に演奏ジャンルを選択する際に使用されるウインドウＷＩＮが表示される。このウインドウＷＩＮは、図２１に図示するように、タイトル文字を構成するオブジェクトＯＢＪＡ－１～ＯＢＪＡ８およびスクロール可能状態を表示するオブジェクトＯＢＪＡ－５０，ＯＢＪＡ－５１が配置されるタイトルバー（オブジェクトＯＢＪＡ－５２）と、８行の色可変のボタン（オブジェクトＯＢＪＡ－５１～ＯＢＪＡ－６０）とから構成されている。

【００５２】各ボタン（オブジェクトＯＢＪＡ－５１～ＯＢＪＡ－６０）では、行当り８ドット×８ドットサイズのオブジェクトＯＢＪＢが１８列配置されている。つまり、オブジェクトＯＢＪＢ－１２８～ＯＢＪＢ－２７１がマトリクス配置されている。これらオブジェクトＯＢＪＢ－１２８～ＯＢＪＢ－２７１は、文字を形成するか、あるいは透明化されるものであり、上記オブジェクトＯＢＪＡ－５１～ＯＢＪＡ－６０より高い表示順位に設定されている。そのため、ボタン上に文字列が描かれているように見える。また、各ボタンは、十字キー１０９の上下キー操作に応じて、前述したカラールックアップテーブル$ＣＬＴ３_{OBJA}$（図１４参照）のカラーブロック番号を替えることで選択状態であるか否かを示すように表示色を変化させ得る。

【００５３】④バンドセーブ画面の構成

図２２に示すバンドセーブ画面においても、上述と同様のウインドウＷＩＮが配置される。バンセーブ画面中のウインドウＷＩＮにおいては、各ボタン上の文字を形成するオブジェクトＯＢＪＢの種類を替えて文字列を書き換えられるようになっている。その際、オブジェクトＯＢＪＢの表示色は、図１５に示したカラールックアップテーブル$ＣＬＴ３_{OBJB}$のカラーブロック番号を＃１（黒）から＃２（ピンク）に替えて書き換え中であることを表示できる。

【００５４】⑤トータルコントロール画面の構成

次に、図２３はトータルコントロール画面の一例を示す図である。この画面では、装置１００の動作諸条件を設定する「ボタン」がオブジェクトＡによって形成される。各ボタンは、図２４（イ）に図示するように、凹状に見えるノーマル状態（ａ）、セレクト状態（ｂ）および凸状に見えるアクティブ状態（ｃ）の３状態となるように構成されている。つまり、図２４（ロ）に示す通り、ボタンの左縁上縁を形成するＯＢＪＡ－Ａ、ボタンの右縁下縁を形成するＯＢＪ－Ｂ、ボタン中央部を形成するＯＢＪＡ－Ｃおよびボタン上の文字を形成するＯＢＪＡ－Ｄ，Ｅによってボタンを形成し、設定操作の内容に応じてこれらオブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃの表示色を替えて上記（ａ）～（ｂ）の状態を表わしている。

【００５５】例えば、図中の「ＲＥＳＵＭＥ」を例として説明する。先ず、未設定状態のボタンでは、上記オブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃにそれぞれカラールックアップテーブル$ＣＬＴ４_{OBJA}$のカラーブロック番号＃０における色データＤ１３，Ｄ１４，Ｄ１５をアサ

インし、凹状に見えるノーマル状態（ａ）にしておく。この状態から十字キー１０９の左右キー操作によりボタン選択すると、オブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃにカラーブロック番号＃８の色データＤ１３，Ｄ１４，Ｄ１５をアサインし、表示色を替えてセレクト状態（ｂ）を表わす。そして、エンターキー１１１を押下して設定を確定させると、オブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃにカラーブロック番号＃７の色データＤ１３，Ｄ１４，Ｄ１５をアサインし、凸状に見えるアクティブ状態（ｃ）にする。このようにすることで、マウス等のポインティングデバイスを備えずとも、あたかもボタンをクリックしたようなＧＵＩ操作環境を実現することが可能になる。

【００５６】Ｃ．実施例の動作

次に、上記構成による実施例の動作について図２５～図４３を参照して説明する。以下では、始めに全体動作の概略を述べた後、本発明の要旨に関わる画像処理の詳細について順次説明して行く。

【００５７】（１）全体動作

(a)概略フロー

本実施例である装置１００の電源スイッチ１０２が投入されると、ＣＰＵ１は図２５に示すフローに従って動作する。すなわち、電源スイッチ１０２を投入すると、ステップＳＡ１に進み、ＲＡＭ３，ＶＲＡＭ６、ＶＤＰ５および音源８の各部に設けられているレジスタやフラグをゼロリセットしたり、初期値をセットする等の初期化を行った後、ステップＳＡ２に進む。ステップＳＡ２では、鍵盤１０１やパネルスイッチ群を走査して押離鍵操作イベント、あるいはスイッチ操作イベントを検出し、検出したイベントに応じて楽音を発生したり、楽音発生する際の各種演奏パラメータをセットする発音処理／音楽処理を行う一方、これら処理に対応した画面表示を発生するための画像処理を実行する。

【００５８】次いで、ステップＳＡ２が完了すると、ＣＰＵ１はステップＳＡ３に処理を進め、オートパワーオフ設定されたか否かを判断する。ここで、パワーオフ状態でないならば、判断結果が「ＮＯ」となり、ステップＳＡ２に処理を戻してメインフロー処理を継続し、一方、パワーオフ設定された時には判断結果が「ＹＥＳ」となって次のステップＳＡ４に処理を進める。ステップＳＡ４では、例えば、装置１００における各部設定状態をメモリ退避するレジューム処理や、発音中の楽音をミュートする消音処理などを行った後に電源オフするパワーオフ処理を実行して動作停止する。

【００５９】(b)メインフロー

ＣＰＵ１の処理が上述のステップＳＡ２に進むと、図２６に示すメインフローを実行してステップＳＢ１に進む。ステップＳＢ１では、鍵盤１０１の押離鍵操作を検出すべく鍵盤スキャンを行う。続いて、ステップＳＢ２では、この鍵盤スキャンにおいて検出した押離鍵操作イベントに基づきキーオン、キーオフ、キーコードおよびベロシティなどの演奏情報を発生すると共に、この演奏情報に従って音源８に楽音発生を指示する発音処理を行う。次いで、ステップＳＢ３では、パネルスイッチ１０３～１１１のスイッチ操作を検出するため、パネルスイッチスキャンを行い、スイッチ操作イベントを発生する。

【００６０】次に、ステップＳＢ４では、生成されたスイッチ操作イベントの内、時系列の自動演奏情報を作成するスイッチ操作を抽出してイベント行列を作成する。このイベント行列とは、自動リズム演奏情報を作成するのであれば、発音タイミングを羅列したリズムパターンとなり、自動演奏情報を作成する時には音高および発音タイミングからなる演奏パターンとなる。次いで、ステップＳＢ５に進むと、ＣＰＵ１は、上記イベント行列として表わされるリズムパターンあるいは演奏パターンを楽音として発生する時に付与する効果形態を設定する音楽処理が行われる。この後、ＣＰＵ１は本発明の要旨に関わる画像処理、すなわち、上述のスイッチ操作イベントに対応した画面を発生すべく表示制御し、続くステップＳＢ７では、その他のシステム処理を施して本メインフローを完了する。

【００６１】（２）画像処理の動作

次に、上記ステップＳＢ６において実行される画像処理の内容について詳述する。ＣＰＵ１の処理が上記ステップＳＢ６に進むと、図２７に示す画像処理ルーチンが起動され、ステップＳＣ１に処理を進める。ステップＳＣ１では、モード切替えがあるか否か、つまり、画面表示を変更するスイッチ操作イベントが発生したかどうかを判断する。ここで、画面表示を変更するスイッチ操作イベントがあれば、判断結果が「ＹＥＳ」となり、次のステップＳＣ２に処理を進め、そうでない場合には後述のステップＳＣ３に処理を進める。ステップＳＣ２に進むと、ＣＰＵ１は切替えるべき画面内容に従い、ＲＯＭ２から対応するオブジェクトコードを抽出してＶＤＰ５内部のオブジェクトメモリ５２にストアする画面切替処理を実行する。なお、画面切替処理の詳細については後述する。

【００６２】次いで、ステップＳＣ３～ＳＣ６では、各画面モード下において発生するスイッチ操作イベントに対応した表示制御を行う。すなわち、ステップＳＣ３にあっては、前述した十字キー１０９やエスケープキー１１０、あるいはエンターキー１１１の操作に応じてバンド画面（図１８参照）中に表示される文字列ＣＨＲ１～ＣＨＲ８の表示態様を変更する。次に、ステップＳＣ４では、十字キー１０９やエンターキー１１１の操作に応じてジャンルセレクト画面（図２０参照）に表示されるウインドウＷＩＮ内の文字列を移動表示したり、色変更する表示制御を行う。そして、ステップＳＣ５では、トータルコントロール画面（図２３参照）における各アイ

テム毎の「ボタン」の表示色を、スイッチ操作に応じて異ならせる表示制御を行う。さらに、ステップＳＣ６では、バンドセーブ画面（図２２参照）で十字キー１０９やエンターキー１１１の操作に応じてウインドウＷＩＮ内の文字列を移動表示したり、色変更する表示制御を行う。

【００６３】これらステップＳＣ３～ＳＣ６においてなされる表示制御の詳細については後述する。また、このステップＳＣ３～ＳＣ６による表示制御を実行するＣＰＵ１は、一定周期毎にＶブランクインタラプト処理ルーチン（後述する）を割込み処理し、表示制御された画面を更新する。すなわち、ステップＳＣ３～ＳＣ６を実行している過程で、ＣＰＵ１はディスプレイＤＰ側の垂直帰線期間に同期してＶブランクインタラプト処理ルーチンを実行して画面表示に必要なオブジェクトコード、画像データおよびカラールックアップテーブルをＶＤＰ５やＶＲＡＭ６側にＤＭＡ転送する。そして、ＶＤＰ５では、こうしてＶＲＡＭ６や自身内部のキャラジェネメモリ５４，オブジェクトメモリ５２にＤＭＡ転送された上記各種データに基づき画面を更新する。

【００６４】(a)画面切替処理ルーチンの動作

次に、画面切替処理ルーチンの動作を各画面毎に説明する。

①初期画面への切替え

上述したように、画面表示を変更するスイッチ操作イベントが発生すると、ＣＰＵ１はステップＳＣ２を介して図２８に示す画面切替処理ルーチンを実行してステップＳＤ１に処理を進める。ステップＳＤ１では、発生したスイッチ操作イベントが初期画面への切替えを指示するものであるか否かを判断する。ここで、初期画面への切替えを指示するイベントであれば、ステップＳＤ１の判断結果が「ＹＥＳ」となり、ＣＰＵ１はステップＳＤ２に処理を進める。ステップＳＤ２では、レジスタＩＧＦに格納される初期画面転送フラグの値を「１」にセットする。

【００６５】次いで、ステップＳＤ３に進むと、ＲＯＭ２に格納されている初期画面用オブジェクトコードＯＢＪ１（図３参照）を、ＲＡＭ３のワークエリアへストアする。このようにすることで、後述するＶブランクインタラプト処理ルーチンにより、垂直帰線消去期間中に初期画面を形成するバックグラウンド画像データ、オブジェクトＡ画像データおよび初期画面用ＣＬＴデータＣＬＴ１がＶＲＡＭ６およびＶＤＰ５側へ転送され、これにより初期画面が更新表示される。

【００６６】②バンド画面への切替え

バンド画面への切替えを指示するスイッチ操作イベントが発生した時には、上記ステップＳＤ１の判断結果が「ＮＯ」となり、ステップＳＤ４に処理を進め、バンド画面への切替えか否かを判断する。そして、この場合、バンド画面への切替えを行うのだから、ステップＳＤ４の判断結果は「ＹＥＳ」となり、次のステップＳＤ５に処理を進める。ステップＳＤ５では、レジスタＢＧＦに格納されるバンド画面転送フラグの値を「１」にセットする。次いで、ステップＳＤ６に進むと、ＲＯＭ２に格納されているバンド画面用オブジェクトコードＯＢＪ２（図３参照）の内からオブジェクトＡ用コードＯＢＪ２$_{OBJA}$を抽出してＲＡＭ３のワークエリアへストアする。

【００６７】続いて、ステップＳＤ７に進むと、ＣＰＵ１は上記ステップＳＤ６と同様に、ＲＯＭ２に格納されているバンド画面用オブジェクトコードＯＢＪ２（図３参照）の内からオブジェクトＢ用コードＯＢＪ２$_{OBJB}$を抽出してＲＡＭ３のワークエリアへストアする。そして、これにより後述のＶブランクインタラプト処理ルーチンにおいて、垂直帰線消去期間中にバンド画面を形成するバンド画面用画像データＩＭ２（図７参照）およびバンド画面用ＣＬＴデータＣＬＴ２（図１２参照）がＶＲＡＭ６およびＶＤＰ５側へ転送されてバンド画面が形成される。

【００６８】③ジャンルセレクト画面への切替え

次に、ジャンルセレクト画面への切替えを指示するスイッチ操作イベントが発生した時には、上記ステップＳＤ１，ＳＤ４の判断結果がいずれも「ＮＯ」となり、ステップＳＤ８に処理を進め、ジャンルセレクト画面への切替えか否かを判断する。そして、この場合、ジャンルセレクト画面への切替えを行うのだから、ステップＳＤ８の判断結果が「ＹＥＳ」となり、次のステップＳＤ９に処理を進める。ステップＳＤ９では、レジスタＪＳＦに格納されるジャンルセレクト画面転送フラグの値を「１」にセットする。次いで、ステップＳＤ１０に進むと、ＲＯＭ２に格納されているジャンルセレクト画面用オブジェクトコードＯＢＪ４（図３参照）の内からオブジェクトＡ用コードＯＢＪ４$_{OBJA}$を抽出してＲＡＭ３のワークエリアへストアする。

【００６９】続いて、ステップＳＤ１１に進むと、ＣＰＵ１は上記ステップＳＤ１０と同様に、ＲＯＭ２に格納されているジャンルセレクト画面用オブジェクトコードＯＢＪ４（図３参照）の内からオブジェクトＢ用コードＯＢＪ４$_{OBJB}$を抽出してＲＡＭ３のワークエリアへストアする。この結果、後述のＶブランクインタラプト処理ルーチンに基づき、垂直帰線消去期間中にジャンルセレクト画面を形成する画像データＩＭ３（図７参照）およびジャンルセレクト画面用ＣＬＴデータＣＬＴ３（図１２参照）がＶＲＡＭ６およびＶＤＰ５側へ転送され、ジャンルセレクト画面が更新される。

【００７０】④トータルコントロール画面への切替え

トータルコントロール画面への切替えを指示するスイッチ操作イベントが発生すると、上述のステップＳＤ１，ＳＤ４，ＳＤ８の判断結果がいずれも「ＮＯ」となり、ステップＳＤ１２に処理を進め、トータルコントロール画面への切替えか否かを判断する。そして、この場合、

トータルコントロール画面への切替えを行うので、ステップＳＤ１２の判断結果は「ＹＥＳ」となり、次のステップＳＤ１３に処理を進める。

【００７１】ステップＳＤ１３では、レジスタＴＣＦに格納されるトータルコントロール画面転送フラグの値を「１」にセットする。次いで、ステップＳＤ１４に進むと、ＲＯＭ２に格納されているトータルコントロール画面用オブジェクトコードＯＢＪ５（図３参照）をＲＡＭ３のワークエリアへストアする。この結果、後述のＶブランクインタラプト処理ルーチンに基づき、垂直帰線消去期間中にトータルコントロール画面を形成する画像データＩＭ４（図７参照）およびトータルコントロール画面用ＣＬＴデータＣＬＴ４（図１２参照）がＶＲＡＭ６およびＶＤＰ５側へ転送されることによって、トータルコントロール画面が更新される。

【００７２】⑤バンドセーブ画面への切替え

バンドセーブ画面への切替えを指示するスイッチ操作イベントが発生すると、上述したステップＳＤ１，ＳＤ４，ＳＤ８，ＳＤ１２の判断結果がいずれも「ＮＯ」となり、ステップＳＤ１５に処理を進め、バンドセーブ画面への切替えか否かを判断する。そして、この場合、バンドセーブ画面への切替えを行うのだから、ステップＳＤ１５の判断結果が「ＹＥＳ」となり、次のステップＳＤ１６に処理を進める。ステップＳＤ１６では、レジスタＢＳＦに格納されるバンドセーブ画面転送フラグの値を「１」にセットする。次いで、ステップＳＤ１７に進むと、ＲＯＭ２に格納されているバンドセーブ画面用オブジェクトコードＯＢＪ６（図３参照）の内からオブジェクトＡ用コード$OBJ6_{OBJA}$を抽出してＲＡＭ３のワークエリアへストアする。

【００７３】続いて、ステップＳＤ１８では、上記ステップＳＤ１７と同様に、ＲＯＭ２に格納されているバンドセーブ画面用オブジェクトコードＯＢＪ６（図３参照）の内からオブジェクトＢ用コード$OBJ6_{OBJB}$を抽出してＲＡＭ３のワークエリアへストアする。この結果、後述のＶブランクインタラプト処理ルーチンに基づき、垂直帰線消去期間中にバンドセーブ画面を形成する画像データＩＭ５（図７参照）およびバンドセーブ画面用ＣＬＴデータＣＬＴ５（図１２参照）がＶＲＡＭ６およびＶＤＰ５側へ転送されてバンドセーブ画面が更新される。なお、画面切替えを指示するイベントとは異なる他のイベントが発生した時には、上述のステップＳＤ１，ＳＤ４，ＳＤ８，ＳＤ１２およびＳＤ１５における各判断結果はいずれも「ＮＯ」となり、ＣＰＵ１はこのルーチンを完了する。

【００７４】(b)バンド画面処理ルーチンの動作

こうして画面切替処理ルーチンが完了した場合、あるいは画面切替え以外のイベントが発生した時、ＣＰＵ１は前述したステップＳＣ３を介して図２９に示すバンド画面処理ルーチンを実行してステップＳＥ１へ処理を進める。ステップＳＥ１では、現在表示中の画面がバンド画面であるかどうかを判断する。ここで、バンド画面を表示するモードでない場合には、判断結果が「ＮＯ」となり、このルーチンを完了する。一方、バンド画面を表示している時には、判断結果が「ＹＥＳ」となり、次のステップＳＥ２に処理を進める。ステップＳＥ２に進むと、ＣＰＵ１は発生したイベントが画像変更を要するイベントであるか否かを判断する。

【００７５】画像変更を要するイベントとは、例えば、バンド画面（図１８参照）における文字列ＣＨＲ１を十字キー１０９の上下キー操作に応じて選択する場合や、各パート毎に表示されるミュージシャンのポートレイト像を入れ替えるバンドメンバー変更操作を指す。そして、画像変更が必要な操作が行われた時には、上記ステップＳＥ２の判断結果が「ＹＥＳ」となり、ステップＳＥ３に処理を進める。一方、画像変更を必要としない操作がなされた時には、判断結果が「ＮＯ」となり、このルーチンを完了する。

【００７６】さて、画像変更が必要な操作が行われた場合、ＣＰＵ１はステップＳＥ３に処理を進め、発生したイベントが文字列を変更するものであるか否かを判断する。ここで、例えば、バンド画面（図１８参照）中の文字列ＣＨＲ１のいずれかを選択すべく十字キー１０９の上キーあるいは下キーを操作してカーソル位置をスクロールさせると、判断結果が「ＹＥＳ」となり、次のステップＳＥ４に処理を進める。ステップＳＥ４では、このスクロール操作に応じて文字列ＣＨＲ１のいずれかを選択する文字列変更処理ルーチン（後述する）を実行する。次いで、ステップＳＥ５では、選択した文字列（オブジェクトＯＢＪＢ）に対応付けられたカラーブロック番号を書き換え、その文字列の表示色を異ならせるカラーブロック変更処理ルーチン（後述する）を実行した後、このルーチンを完了する。

【００７７】一方、上記ステップＳＥ３において判断結果が「ＮＯ」の場合、すなわち、文字列変更以外の操作がなされた時には、ステップＳＥ６に処理を進める。ステップＳＥ６では、為された操作がバンドメンバーを変更する操作、つまり、バンド画面中に表示されるミュージシャンのポートレイト像のいずれかを他のものと入れ替えるバンドメンバー変更操作であるか否かを判断する。そして、その操作が行われると、ここでの判断結果が「ＹＥＳ」となり、次のステップＳＥ７に処理を進める。ステップＳＥ７では、変更指定されたパートのポートレイト像を、指定座標位置に表示すると共に、変更対象とされたポートレイト像の下部に配置される「パート名」部分の表示色を異ならせるバンドメンバー変更処理ルーチン（後述する）を実行して本ルーチンを完了する。

【００７８】(c)文字列変更処理ルーチンの動作

上述したように、バンド画面中の文字列ＣＨＲ１のいず

れかを選択すべく十字キー１０９の上キーあるいは下キーを操作してカーソル位置をスクロールさせると、ＣＰＵ１はステップＳＥ４を介して図３０に示す文字列変更処理ルーチンを実行してステップＳＦ１に処理を進める。ステップＳＦ１では、操作されたスクロールの方向を判別する。スクロール方向が「上」、つまり、十字キー１０９の上キーが操作された時にはステップＳＦ２に進み、ＲＯＭ２上の表示エリアを示すＹ方向アドレスを１行分デクリメントする。

【００７９】すなわち、バンド画面中において移動表示可能な文字を形成するオブジェクトＢ用コードＯＢＪ２$_{OBJB}$は、ＲＯＭ２のオブジェクト記憶エリアＥ$_{OBJ}$において、画面上の表示形態に合致した形で格納されている。その格納エリアはＸ方向（列方向）１８文字列、Ｙ方向（行方向）６行となっており、これが文字列ＣＨＲ１として画面表示される訳である。したがって、十字キー１０９の上キーが操作されて画面上の上側へスクロールした場合には、上記格納エリアにおけるＯＢＪＢコードのＹ方向読み出しアドレスを１デクリメントする。また、これと同様、十字キー１０９の下キーが操作されて画面上の下側へスクロールした時には、上記ステップＳＦ１の判別に基づき、ステップＳＦ３に処理を進め、この場合、Ｙ方向読み出しアドレスを１インクリメントする。

【００８０】次いで、ステップＳＦ４に進むと、ＣＰＵ１はスクロール操作に応じて更新された現アドレス（Ｘ，Ｙ）を読み出し開始ポインタとする。そして、次のステップＳＦ５では、ポインタレジスタｉに「１」をセットする。なお、このポインタレジスタｉの値は、後述するように順次インクリメントされて読み出し開始ポインタを基準とする相対アドレスとして扱われる。そして、ステップＳＦ６以降では、スクロールされたカーソル位置を基準にして上記格納エリアから１８文字列×６行分のＯＢＪＢコードをポインタレジスタｉの値に応じて読み出す。すなわち、ステップＳＦ６では、１８文字列、６行分のＯＢＪＢコードを読み出したか否かを判断する。ここで、１８文字列、６行分のＯＢＪＢコードを読み出した場合には、判断結果が「ＹＥＳ」となり、ステップＳＦ７に進む。ステップＳＦ７では、読み出し開始ポインタをリセットして本ルーチンを完了する。

【００８１】一方、ＯＢＪＢコードの読み出しが完了していない時には、上記ステップＳＦ６の判断結果は「ＮＯ」となり、ステップＳＦ８に進み、ポインタレジスタｉの値と現アドレス（Ｘ，Ｙ）とに基づいて指定される文字列を形成するＯＢＪＢコード列を読み込む。次いで、ステップＳＦ９では、読み込んだＯＢＪＢコード列を一文字づつ解釈し、解釈したＯＢＪＢコードが「スペース（空白）」であるか否かを判断する。ここで、「スペース（空白）」であると、判断結果が「ＹＥＳ」となり、次のステップＳＦ１０に処理を進め、透明キャラクタ（透明色指定）に設定する。

【００８２】これに対し、「スペース（空白）」でない時には、上記ステップＳＦ９の判断結果が「ＮＯ」となり、ステップＳＦ１１に処理を進め、そのＯＢＪＢコードが指定するオブジェクトＢキャラクタ（文字画像）に設定する。そして、この後、ＣＰＵ１はステップＳＦ１２に処理を進め、解釈したＯＢＪＢコードをＶＤＰ５内部のオブジェクトメモリ５２にストアし、続くステップＳＦ１３においてポインタレジスタｉを１インクリメントして上述のステップＳＦ６に処理を戻す。

【００８３】このように、文字列変更処理では、十字キー１０９の操作に応じてスクロールされたカーソル位置を基準にしてＲＯＭ２の格納エリアから１８文字列、６行分のＯＢＪＢコードを、ポインタレジスタｉの値に応じて読み出し、読み出したＯＢＪＢコードが「スペース（空白）」である時には透明キャラクタ（透明色指定）を設定し、そうでない時にはそのＯＢＪＢコードが指定するオブジェクトＢキャラクタ（文字画像）に設定するようにしている。つまり、表示変更される文字を全てオブジェクトＢ画像に設定しておき、表示変更する際には対応するＯＢＪＢコードを指定し、消去表示の際には透明色に指定すれば良い為、キャラクタ（文字）毎に表示形態と表示座標位置とを指定する従来のものに比べて高速な画面表示と成り得る。

【００８４】(d)カラーブロック変更処理（ＯＢＪＢ）ルーチンの動作

次に、表示変更の対象となるオブジェクトＯＢＪＢの表示色を変更するカラーブロック変更処理ルーチンの動作について図３１を参照して説明する。オブジェクトＢ画像は、前述したように、ドット当り１ビットのカラーコードが割り当てられているため、例えば、図１５に図示するカラールックアップテーブルＣＬＴ３$_{OBJB}$から判るように、カラーブロック番号を替えて、表示色を「黒」と「ピンク」とに交互に切替可能となっている。オブジェクトＯＢＪＢのカラーブロックを変更する場合、「黒」が後述の「ノーマル色」に相当し、「ピンク」が後述の「アクティブ色」に相当する。

【００８５】いま、例えば、上述の文字列変更処理ルーチンによって文字列（オブジェクトＯＢＪＢ）の変更が完了したとする。そうすると、ＣＰＵ１はこの変更された文字列の表示色を替えるべく、前述したステップＳＥ５を介して図３１に示すカラーブロック変更処理（ＯＢＪＢ）ルーチンを実行してステップＳＧ１に処理を進める。ステップＳＧ１では、先ず変更対象とされるオブジェクトを指定する。次いで、ステップＳＧ２では、その対象とされているオブジェクトに割り当てられている表示色が「ノーマル色」であるかどうかを判断する。そして、「ノーマル色」であれば、判断結果が「ＹＥＳ」となり、次のステップＳＧ３に処理を進め、そのオブジェクトに「アクティブ色」を割り付けるカラーブロック番

号を算出する。

【００８６】一方、対象となるオブジェクトに割り当てられている表示色が「アクティブ色」であった時には、上記ステップＳＧ２の判断結果が「ＮＯ」となり、ＣＰＵ１はステップＳＧ４に処理を進め、そのオブジェクトに「ノーマル色」を割り付けるカラーブロック番号を算出する。そして、この後、ステップＳＧ５に進むと、対象となるオブジェクトＯＢＪＢコード中のカラーブロック番号を、上記ステップＳＧ３、あるいはステップＳＧ４にて算出したカラーブロック番号に書き替える。このように、カラーブロック変更処理（ＯＢＪＢ）では、ノーマル状態で表示変更する時にはアクティブ色のカラーブロック番号を割り当て、その反対の場合にはノーマル色のカラーブロック番号を割り当てる。

【００８７】(e)バンドメンバー変更処理ルーチンの動作

バンド画面において、画面中に表示されるミュージシャンの５種類のポートレイト像のいずれかを他のものと入れ替えるバンドメンバー変更操作がなされると、ＣＰＵ１は前述のステップＳＥ７（図２９参照）を介して図３２に示すバンドメンバー変更処理ルーチンを実行してステップＳＨ１に進む。ステップＳＨ１では、オブジェクト移動処理ルーチン（後述する）を実行してメンバーのポートレイト像を指定位置に移動表示させる。そして、ステップＳＨ２では、オブジェクト色変更処理ルーチン（後述する）の動作に基づき、この移動表示されたポートレイト像に対応するオブジェクトＯＢＪＡの表示色を変更する。

【００８８】(f)オブジェクト移動処理ルーチンの動作

上記ステップＳＨ１が実行されると、ＣＰＵ１は図３３に示すオブジェクト移動処理ルーチンを実行してステップＳＪ１に処理を進める。ステップＳＪ１では、選択メンバーを表示する操作がなされたか否かを判断する。ここで、選択メンバーを表示する操作とは、図１８に図示するバンド画面の下部に表示される文字列ＣＨＲ７（「ＭＥＭＢＥＲ」）の位置にカーソルを移動させ、そこでエンターキー１１１を押下してバンドメンバー変更モードに遷移させた後、変更すべき「パート」を指定する操作を指す。このような操作がなされた時、上記ステップＳＪ１の判断結果が「ＹＥＳ」となり、次のステップＳＪ２に処理を進める。一方、選択メンバーを表示する操作がなされない場合には、後述のステップＳＪ４へ進む。

【００８９】ステップＳＪ２では、選択項目表示エリアを形成する各オブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を、画面表示範囲外となる表示領域Ｂへ移動するよう変更する。なお、ここで言う選択項目表示エリアとは、バンド画面中のポートレイト像より下半分の画面範囲を指す。そして、次のステップＳＪ３では、選択されたメンバーに対するオブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を、上記の選択項目表示エリアの替わりとして表示領域Ａ側へ移動する。なお、選択されたメンバーとは、メンバー変更の対象となる「パート」に予め割り当てられた選択候補のポートレイト像を言う。つまり、上記ステップＳＪ２～ＳＪ３では、選択メンバーを表示する操作がなされると、選択項目表示エリアを画面表示範囲外となる表示領域Ｂへ移動させる一方、表示領域Ｂ側に配置されていた選択候補のポートレイト像を、選択項目表示エリアの替わりとして表示領域Ａ側へ移動する。これにより、図１９に図示したバンドメンバー変更画面が形成される。

【００９０】次に、ステップＳＪ４に進むと、ＣＰＵ１は選択メンバーを変更する操作、つまり、新たな候補となるポートレイト像を表示させる操作が行われたかどうかを判断する。そして、選択メンバーを変更する操作がなされると、判断結果が「ＹＥＳ」となり、次のステップＳＪ５に処理を進め、変更により選択されたメンバーのオブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を表示領域Ａ側に移動して新たな候補のポートレイト像として表示する。次いで、ステップＳＪ６では、新たな候補のポートレイト像が表示されることに呼応して、変更により外れたメンバーのオブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を表示領域Ｂ側に移動して非表示とする。

【００９１】上記ステップＳＪ５，ＳＪ６の動作を図１９に図示するバンドメンバー変更画面を例に挙げて説明する。まず画面下半分側に選択候補となる３人のポートレイト像が表示されているとする。そして、この状態から新たな候補となるポートレイト像を表示させるには、選択メンバーを変更する操作を行う。具体的には、装置パネル面に配設される十字キー１０９の右キーあるいは左キーを操作して候補を入れ替える。つまり、右キーを操作した時には、右端のポートレイト像が表示領域Ｂ側へ移動して非表示になり、これに応じて中央および左端のポートレイト像が順次右側へシフトする。これにより、左端に空きができ、この位置に新たな候補のポートレイト像が表示領域Ｂ側から移動して表示される。なお、左キーを操作した時には、上述とは逆に中央、右端のポートレイト像がそれぞれ左シフトし、右端位置に新たな候補が表示される。したがって、左右キーのいずれかを操作し続ければ、左シフト巡回あるいは右シフト巡回して候補のポートレイト像が順次更新される。

【００９２】以上のようにして選択メンバーの変更がなされると、ＣＰＵ１は図３４に示すステップＳＪ７に進み、メンバー決定する操作が行われたか否かを判断する。ここで、所定のメンバー（ポートレイト像）を決定すべくエンターキー１１１を押下すると、判断結果が「ＹＥＳ」となり、次のステップＳＪ８に処理を進める。ステップＳＪ８では、現在決定したメンバーのオブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）と、

変更対象メンバーのオブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）とを相互に入れ替える。これにより、現在決定したメンバーのポートレイト像が表示領域Ａ側に固定される。

【００９３】次いで、ステップＳＪ９に進むと、表示状態を元に戻すエスケープキー１１０が操作されたか否かを判断し、当該キー１１０が操作されない時には判断結果が「ＮＯ」となり、一旦、本ルーチンを完了する。一方、表示状態を元に戻すべく、エスケープキー１１０が操作された時には、判断結果が「ＹＥＳ」となり、次のステップＳＪ１０に処理を進める。ステップＳＪ１０では、選択候補として表示されているポートレイト像の各オブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を、画面表示範囲外となる表示領域Ｂへ移動するよう変更する。そして、次のステップＳＪ１１では、画面表示範囲外となる表示領域Ｂへ退避しておいた選択項目表示エリアの各オブジェクトＯＢＪＡコードの表示位置座標（Ｘ，Ｙ）を、表示領域Ａ側の元に位置へ表示するよう変更する。つまり、上記ステップＳＪ１０～ＳＪ１１では、表示状態を元に戻すエスケープキー操作がなされると、選択候補として表示されているポートレイト像を画面表示範囲外となる表示領域Ｂへ戻す一方、表示領域Ｂ側に退避しておいた選択項目表示エリアの各オブジェクトを元の表示領域Ａ側へ移動して表示状態に設定する。これにより、バンドメンバー変更画面からバンド画面へ切替わる。

【００９４】(g)オブジェクト色変更処理ルーチンの動作

こうしてバンドメンバーの変更に応じてポートレイト像の表示位置を替えるオブジェクト移動処理ルーチンが行われると、ＣＰＵ１は前述したステップＳＨ２（図３２参照）を介して図３５に示すオブジェクト色変更処理ルーチンを実行してステップＳＫ１に処理を進める。ステップＳＫ１では、上述したオブジェクト移動処理ルーチンの処理に連携して表示色を変更すべきオブジェクトを求める。そして、ステップＳＫ２以降、その色変更の対象となるオブジェクトの表示色に応じてオブジェクトコード中のカラーブロック番号を書き替える。以下、色変更の内容別に処理を説明する。

【００９５】①ノーマル色に変更する場合

バンドメンバー変更操作を完了してからバンド画面へ復帰する際、処理対象のオブジェクトＯＢＪＡは定常状態を表わすノーマル色に色変更される。この場合、ステップＳＫ２の判断結果が「ＹＥＳ」となり、ステップＳＫ３に進み、対象となるオブジェクトＯＢＪＡのノーマル色を持つカラーブロック番号を算出する。そして、次のステップＳＫ４において、対象となるオブジェクトＯＢＪＡコード中のカラーブロック番号を、ノーマル色のカラーブロック番号に書き替える。

【００９６】②セレクト色に変更する場合

バンドメンバー変更時に所望のメンバーを選択する際、処理対象のオブジェクトＯＢＪＡは選択状態を表わすセレクト色に色変更される。この場合、ステップＳＫ５の判断結果が「ＹＥＳ」となり、ステップＳＫ６に進み、対象となるオブジェクトＯＢＪＡのセレクト色を持つカラーブロック番号を算出する。そして、次のステップＳＫ７において、対象となるオブジェクトＯＢＪＡコード中のカラーブロック番号を、セレクト色のカラーブロック番号に書き替える。

【００９７】③アクティブ色に変更する場合

バンドメンバー変更時に所望のメンバーを決定する際、処理対象のオブジェクトＯＢＪＡは決定状態を表わすアクティブ色に色変更される。この場合、図３６に示すステップＳＫ８の判断結果が「ＹＥＳ」となり、ステップＳＫ９に進み、対象となるオブジェクトＯＢＪＡのアクティブ色を持つカラーブロック番号を算出する。そして、次のステップＳＫ７において、対象となるオブジェクトＯＢＪＡコード中のカラーブロック番号を、アクティブ色のカラーブロック番号に書き替える。

【００９８】このように、オブジェクト色変更処理ルーチンでは、前述したオブジェクト移動処理ルーチンの処理に連携して表示色を変更すべきオブジェクトを求め、求めたオブジェクトの遷移状態に従って、対応するＯＢＪＡコード中のカラーブロック番号を、「ノーマル色」、「セレクト色」および「アクティブ色」のいずれかを示すものに書き換える。したがって、例えば、色変更するオブジェクトとして、ポートレイト像の枠部分や、ポートレイト像の下部に配置されるパート名称部分を、遷移状態に応じて表示色を異ならせている。これにより、マウスを持たない装置にあっても、所謂、ＧＵＩ環境が実現でき、操作性を向上することが可能になる。

【００９９】(h)ジャンルセレクト画面処理ルーチンの動作

以上のようにして、バンド画面処理ルーチンが完了すると、ＣＰＵ１は前述したステップＳＣ４（図２７参照）を介して図３７に示すジャンルセレクト画面処理ルーチンを実行してステップＳＬ１へ処理を進める。ステップＳＬ１では、現在のモードが演奏ジャンルを選択するジャンルセレクトモード下にあるかどうかを判断し、そのモードでない時には、ここでの判断結果が「ＮＯ」となり、このルーチンを完了する。一方、ジャンルセレクトモード下にあれば、判断結果が「ＹＥＳ」となり、次のステップＳＬ２に処理を進める。ステップＳＬ２では、発生したイベントが画像変更を要するイベントであるか否かを判断する。

【０１００】ジャンルセレクト画面において画像変更を要するイベントとは、例えば、ジャンルセレクト画面（図２０参照）におけるウインドウＷＩＮ中の文字列を、十字キー１０９の上下キー操作に応じて選択する操作や、この選択した文字列部分に新たに文字入力して文

字列を変更する操作を指す。そして、上記の選択操作あるいは文字列変更操作がなされた場合、ステップＳＬ２の判断結果が「ＹＥＳ」となり、ステップＳＬ３に処理を進める。なお、画像変更を必要としない操作であると、ここでの判断結果が「ＮＯ」となり、このルーチンを完了する。

【０１０１】画像変更を要するイベントが生成され、ステップＳＬ３に処理を進めた場合、ＣＰＵ１はその操作が文字列変更するものであるか否かを判断する。ここで、ウインドウＷＩＮ中の文字列を十字キー１０９の上下キー操作に応じて選択するだけの場合には、判断結果が「ＮＯ」となり、ステップＳＬ４に処理を進める。ステップＳＬ４では、十字キー１０９の操作に応じてウインドウＷＩＮ内を移動するカーソル位置にある文字（オブジェクトＯＢＪＢ）の表示色を異ならせるカラーブロック変更処理（ＯＢＪＢ）ルーチン（図３１参照）を実行する。

【０１０２】すなわち、前述したように、カラーブロック変更処理（ＯＢＪＢ）ルーチンでは、ノーマル状態（黒）にある「文字」部分にカーソルが位置し、表示変更指定された時には、その「文字」をアクティブ色（ピンク）で表示するよう対応するオブジェクトコードのカラーブロック番号を書き替え、これとは逆に、アクティブ状態からノーマル状態に表示変更指定された時には、ノーマル色のカラーブロック番号を割り当てる。

【０１０３】これに対し、ウインドウＷＩＮ中の文字列を、十字キー１０９の上下キー操作に応じて選択した文字列に新たな文字を入力して文字列を変更する場合、上述のステップＳＬ３の判断結果が「ＹＥＳ」となり、ＣＰＵ１はステップＳＬ５に処理を進める。ステップＳＬ５では、前述した文字列変更処理ルーチン（図３０参照）を実行する。すなわち、ジャンルセレクト画面において文字列変更を行う場合、十字キー１０９の操作に応じてスクロールされたカーソル位置を基準にしてＲＯＭ２の格納エリアから１８文字列、８行分のＯＢＪＢコードをポインタレジスタｉの値に応じて読み出し、読み出したＯＢＪＢコードが「スペース（空白）」である時には透明キャラクタ（透明色指定）を設定し、そうでない時にはそのＯＢＪＢコードが指定するオブジェクトＢキャラクタ（文字画像）に設定し、本ルーチンを完了する。

【０１０４】(i)トータルコントロール画面処理ルーチンの動作

次に、トータルコントロール画面処理ルーチンの動作について図３８～図４０を参照して説明する。上述のジャンルセレクト画面処理ルーチンが完了すると、ＣＰＵ１はステップＳＣ５（図２７参照）を介して図３８に示すトータルコントロール画面処理ルーチンを実行してステップＳＭ１へ処理を進める。ステップＳＭ１では、現在のモードが装置全体の動作態様を設定するトータルコントロールモード下であるか否かを判断し、そのモードでない時には判断結果が「ＮＯ」となり、このルーチンを完了する。

【０１０５】一方、トータルコントロールモード下にあれば、判断結果が「ＹＥＳ」となり、次のステップＳＭ２に処理を進める。ステップＳＭ２では、発生したイベントがトータルコントロール画面内の画像を変更するイベントであるか否かを判断する。ここで、画像を変更するイベントとは、例えば、トータルコントロール画面（図２３参照）に表示される「ボタン」を、十字キー１０９の操作に応じて選択する操作や、この選択した「ボタン」に対応付けられている動作態様を設定する際になされるエンターキー１１１の押下操作を指す。そして、十字キー１０９あるいはエンターキー１１１が操作されると、ステップＳＭ２の判断結果が「ＹＥＳ」となり、ステップＳＭ３に処理を進める。なお、画像変更を必要としない操作であると、ここでの判断結果が「ＮＯ」となり、このルーチンを完了する。

【０１０６】さて、十字キー１０９あるいはエンターキー１１１が操作され、画像変更を要するイベントが発生した場合、ＣＰＵ１はステップＳＭ３に処理を進め、その操作が「ボタン」の表示状態を替えるものであるか否かを判断する。ここで、「ボタン」の表示状態が変化しないイベントである時には、判断結果が「ＮＯ」となり、次のステップＳＭ４に処理を進め、例えば、装置全体の設定を初期化するイニシャライズ等、その他の変更処理を実行して本ルーチンを完了する。一方、「ボタン」の表示状態を替えるイベントが発生した時には、上記ステップＳＭ３の判断結果が「ＹＥＳ」となり、ＣＰＵ１はステップＳＭ５に処理を進め、後述のボタン用カラーブロック変更処理ルーチンを実行し、該当ボタンの表示状態を、図２４に図示した”凹状に見えるノーマル状態”、”セレクト状態”および”凸状に見えるアクティブ状態”のいずれかにすべく、「ボタン」を形成するオブジェクトの表示色を替える。

【０１０７】(j)ボタン用カラーブロック変更処理ルーチンの動作

上述したように、「ボタン」の表示状態を替えるイベントが発生すると、ＣＰＵ１はステップＳＭ５を介して図３９に示すボタン用カラーブロック変更処理ルーチンを実行する。なお、このルーチンでは、説明の簡略化を図るため、各種ボタンの内、「ＢｅｅｐＯｎボタン」に関するカラーブロック変更を一例に挙げて説明して行く。本ルーチンが実行されると、ＣＰＵ１はステップＳＮ１に処理を進め、処理対象が「ＢｅｅｐＯｎボタン」であるかどうかを判断する。「ＢｅｅｐＯｎボタン」以外であれば、判断結果が「ＮＯ」となり、ステップＳＮ２に進む。そして、ステップＳＮ２では、「ＢｅｅｐＯｎボタン」以外のその他のボタンに関するカラーブロック変更処理を行う。一方、処理対象が「ＢｅｅｐＯｎボタ

ン」の時には、ステップＳＮ１の判断結果が「ＹＥＳ」となり、後述のＢｅｅｐＯｎボタンカラーブロック変更処理ルーチンを実行し、当該ボタンについてなされる操作形態に応じて「ボタン」形成オブジェクトの表示色を変更してボタン表示態様を異ならせる。

【０１０８】(k)ＢｅｅｐＯｎボタンカラーブロック変更処理ルーチンの動作

本ルーチンが実行されると、ＣＰＵ１は図４０に示すステップＳＰ１に処理を進め、ＢｅｅｐＯｎボタンについてなされる操作形態に応じて目的の表示色を設定する。以下、操作形態別に説明する。

①ボタン選択時

十字キー１０９を操作してＯＮボタン、あるいはＯＦＦボタンのいずれかにカーソルを位置させることでボタン選択がなされ、この場合、目的の表示色は選択状態を表わす「セレクト色」となり、ステップＳＰ２に処理を進める。ステップＳＰ２では、この「セレクト色」に対応するカラーブロック番号を算出し、レジスタdestCblockにストアする。図１６に示すカラールックアップテーブル$CLT4_{OBJA}$の場合、「セレクト色」はカラーブロック＃８なので、レジスタdestCblockに「８」がセットされる。

【０１０９】②ボタン決定（設定）時

上記ボタン選択状態においてエンターキー１１１をオン操作することでボタン決定がなされ、この場合、目的の表示色は決定（設定）状態を表わす「アクティブ色」となり、ステップＳＰ３に処理を進めるステップＳＰ３では、この「アクティブ色」に対応するカラーブロック番号を算出し、レジスタdestCblockにストアする。図１６に示すカラールックアップテーブル$CLT4_{OBJA}$の場合、「アクティブ色」はカラーブロック＃７なので、レジスタdestCblockに「７」がセットされる。

【０１１０】③設定取消時

ボタン決定時にエンターキー１１１がオン操作された場合、その操作がなされたボタンと排他的関係にある対のボタンの設定を取り消して未設定状態を表わす「ノーマル色」となり、ステップＳＰ４に処理を進める。ステップＳＰ４では、この「ノーマル色」に対応するカラーブロック番号を算出し、レジスタdestCblockにストアする。図１６に示すカラールックアップテーブル$CLT4_{OBJA}$の場合、「ノーマル色」はカラーブロック＃１なので、レジスタdestCblockに「１」がセットされる。

【０１１１】こうして表示態様に応じたカラーブロック番号がレジスタdestCblockにセットされると、ＣＰＵ１はステップＳＰ５に処理を進める。ステップＳＰ５では、変更対象ボタン（ＯＮボタンあるいはＯＦＦボタン）を構成するオブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃ（図２４（ロ）参照）の内、色変更すべきオブジェクトを抽出する。次いで、ステップＳＰ６では、抽出したオブジェクトに対応するＯＢＪＡコードの上位４ビット

に割り当てられるカラーブロック値を、レジスタdestCblockに格納されているカラーブロック番号に書き替える。この結果、ノーマル色にされた時にはボタンが凹状に、アクティブ色にされた時には凸状に見えるように表示態様が変更され、さらに、セレクト色の時には両者の中間的な状態を表わすように表示態様が変更される。

【０１１２】(l)バンドセーブ画面処理ルーチンの動作

上述のトータルコントロール画面処理ルーチンが完了すると、ＣＰＵ１はステップＳＣ６（図２７参照）を介して図４１に示すバンドセーブ画面処理ルーチンを実行してステップＳＱ１へ処理を進める。ステップＳＱ１では、現在のモードがバンド登録するバンドセーブモード下であるか否かを判断し、そのモードでない時には判断結果が「ＮＯ」となり、このルーチンを完了する。

【０１１３】一方、バンドセーブモード下にあれば、判断結果が「ＹＥＳ」となり、次のステップＳＱ２に処理を進める。ステップＳＱ２では、発生したイベントがバンドセーブ画面内の画像を変更するイベントであるか否かを判断する。なお、画像を変更するイベントとは、例えば、バンドセーブ画面（図２２参照）に表示されるウインドウＷＩＮ内の文字列を、十字キー１０９の操作に応じて選択する操作や、選択した文字列を書き換える文字列変更操作を指す。そして、これら操作がなされた場合、ステップＳＱ２の判断結果が「ＹＥＳ」となり、ステップＳＱ３に処理を進め、それ以外の操作が行われた時には、画像変更を必要としない操作であるとして判断結果が「ＮＯ」となり、本ルーチンを完了する。

【０１１４】さて、十字キー１０９あるいはエンターキー１１１が操作され、画像変更を要するイベントが発生したとする。そうすると、ＣＰＵ１はステップＳＱ３に処理を進め、その操作が「文字列選択」するものであるか否かを判断する。ここで、十字キー１０９の上下キー操作により「文字列選択」が行われた時には、判断結果が「ＹＥＳ」となり、ステップＳＱ４に処理を進める。ステップＳＱ４では、後述のカラーブロック変更処理（ＯＢＪＡ）ルーチンを実行し、選択された文字列（オブジェクトＯＢＪＢ）と同じ位置に置かれるオブジェクトＯＢＪＡの表示色を「セレクト色」に変更して選択状態を示し、一旦、本ルーチンを完了する。

【０１１５】一方、「文字列選択」以外の操作が行われた時には、上記ステップＳＱ３の判断結果が「ＮＯ」となり、ステップＳＱ５に処理を進める。ステップＳＱ５では、選択した文字列を書き換える操作が行われたか否かを判断し、そうした操作がなされない場合には、判断結果を「ＮＯ」としてステップＳＱ６に処理を進め、その操作に対応した処理を実行する。これに対し、選択した文字列を書き換える操作が行われた場合には、ステップＳＱ５の判断結果が「ＹＥＳ」となり、ステップＳＱ７に処理を進める。

【０１１６】ステップＳＱ７に進むと、ＣＰＵ１は前述

した文字列変更処理ルーチン（図３０参照）を実行する。この文字列変更処理ルーチンでは、十字キー１０９の操作に応じてスクロールされたカーソル位置を基準にしてＲＯＭ２の格納エリアから１８文字列、８行分のＯＢＪＢコードをポインタレジスタｉの値に応じて読み出し、読み出したＯＢＪＢコードが「スペース（空白）」である時には透明キャラクタ（透明色指定）を設定し、そうでない時にはそのＯＢＪＢコードが指定するオブジェクトＢキャラクタ（文字画像）に設定する。そして、ステップＳＱ８に進むと、ＣＰＵ１は前述したカラーブロック変更処理（ＯＢＪＢ）ルーチン（図３１参照）を実行し、対象オブジェクトがノーマル色の時にはアクティブ色に色変更させ、これとは逆の時にはノーマル色に色変更させる。

【０１１７】(m)カラーブロック変更処理（ＯＢＪＡ）ルーチンの動作

上述のステップＳＱ４を介して図４２に示すカラーブロック変更処理（ＯＢＪＡ）ルーチンが実行されると、ＣＰＵ１はステップＳＲ１に処理を進める。先ず、ステップＳＲ１では、十字キー１０９の操作に応じて移動するカーソル位置と合致する対象オブジェクトを求める。そして、ステップＳＲ２以降、その色変更の対象となるオブジェクトの表示色に応じてオブジェクトコード中のカラーブロック番号を書き替える。以下、色変更の内容別に処理を説明する。

【０１１８】①ノーマル色に変更する場合

十字キー１０９の操作により選択対象から外れたオブジェクトＯＢＪＡは定常状態を表わすノーマル色に色変更される。この場合、ステップＳＲ２の判断結果が「ＹＥＳ」となり、ステップＳＲ３に進み、対象となるオブジェクトＯＢＪＡのノーマル色を持つカラーブロック番号を算出する。そして、次のステップＳＲ４において、対象となるオブジェクトＯＢＪＡコード中のカラーブロック番号を、ノーマル色のカラーブロック番号に書き替える。

【０１１９】②セレクト色に変更する場合

十字キー１０９の操作により選択対象とされたオブジェクトＯＢＪＡは選択状態を表わすセレクト色に色変更される。この場合、ステップＳＲ５の判断結果が「ＹＥＳ」となり、ステップＳＲ６に進み、対象となるオブジェクトＯＢＪＡのセレクト色を持つカラーブロック番号を算出する。そして、次のステップＳＲ７において、対象となるオブジェクトＯＢＪＡコード中のカラーブロック番号を、セレクト色のカラーブロック番号に書き替える。

【０１２０】このように、カラーブロック変更処理（ＯＢＪＡ）ルーチンでは、十字キー１０９の操作により選択対象とされたオブジェクトＯＢＪＡの表示色を「セレクト色」に変更して選択状態を示す一方、選択対象から外れたオブジェクトＯＢＪＡを「ノーマル色」に戻すようにしている。

【０１２１】(n)Ｖブランクインタラプト処理ルーチンの動作

さて、ＣＰＵ１はこれまで説明した各ルーチンを実行している過程で、ディスプレイＤＰ側の垂直帰線期間に同期して画面表示に必要なオブジェクトコード、画像データおよびカラールックアップテーブルをＶＤＰ５やＶＲＡＭ６側にＤＭＡ転送するＶブランクインタラプト処理ルーチンを実行している。本ルーチンは、垂直帰線期間に入る毎に割込み実行され、その時の画面モードに対応したオブジェクトコード、画像データおよびカラールックアップテーブルをＶＤＰ５やＶＲＡＭ６側にＤＭＡ転送するものであり、以下では各画面モード別に動作説明する。

【０１２２】①初期画面モード

まず、垂直帰線期間に入る毎に本ルーチンが割込み実行されると、ＣＰＵ１は図４３に示すステップＳＳ１に処理を進める。ステップＳＳ１では、現在の画面モードが初期画面モードであるか否かを判断する。そして、初期画面モードであると、判断結果が「ＹＥＳ」となり、次のステップＳＳ２に処理を進める。ステップＳＳ２では、レジスタＩＧＦに格納される初期画面転送フラグが「１」であるか否かを判断する。初期画面転送フラグは、前述した画面切替処理ルーチン（図２８参照）において初期画面に切替えられた時点で「１」がセットされるので、初期画面に切替えられている時には、判断結果が「ＹＥＳ」となり、次のステップＳＳ３に処理を進める。

【０１２３】ステップＳＳ３では、初期画面を形成するバックグラウンド画像データ$IM1_{BG}$，オブジェクトＡ画像データ$IM1_{OBJA}$（図７参照）をＲＯＭ２からＶＲＡＭ６へＤＭＡ転送する。次いで、ステップＳＳ４に進むと、上記画像データ$IM1_{BG}$，$IM1_{OBJA}$に対応付けられたカラールックアップテーブル$CLT1_{BG}$，$CLT1_{OBJA}$をＲＯＭ２からカラールックアップテーブル部６５（図１７参照）のオブジェクトＡ用テーブルエリアＣＬＴＡにＤＭＡ転送する。そして、ステップＳＳ５に進み、レジスタＩＧＦに格納される初期画面転送フラグをゼロリセットして転送完了を表わし、ステップＳＳ６に処理を進める。

【０１２４】ステップＳＳ６では、前述した画面切替処理ルーチンによりＲＡＭ３に格納された初期画面用ＯＢＪＡコード$OBJ1_{OBJA}$を読み出してＶＤＰ５内部のオブジェクトメモリ５２へ転送する。この結果、ＶＤＰ５は次フレームで表示すべき初期画面を発生する。なお、上述のステップＳＳ２の判断結果が「ＮＯ」の場合、すなわち、既に転送が完了してレジスタＩＧＦに格納される初期画面転送フラグがゼロリセットされている時には、上記ステップＳＳ６を実行して初期画面用ＯＢＪＡコード$OBJ1_{OBJA}$をオブジェクトメモリ（ＯＭ）５２

へ転送する。

【0125】②バンド画面モード

この場合、ステップSS7の判断結果が「YES」となり、ステップSS8に処理を進める。ステップSS8では、レジスタBGFに格納されるバンド画面転送フラグが「1」であるか否かを判断する。バンド画面転送フラグは、画面切替処理ルーチン（図28参照）においてバンド画面に切替えられた時点で「1」がセットされるから、バンド画面に切替えられている時には判断結果が「YES」となり、次のステップSS9に処理を進める。ステップSS9では、バンド画面を形成するバックグラウンド画像データ$IM2_{BG}$、オブジェクトA画像データ$IM2_{OBJA}$（図7参照）をそれぞれROM2からVRAM6へDMA転送する。続いて、ステップSS10では、バンド画面を形成するオブジェクトB画像データ$IM2_{OBJB}$をROM2からVDP5内部のキャラジェネメモリ（CGM）54へDMA転送する。

【0126】次いで、ステップSS11に進むと、CPU1は上記画像データ$IM2_{BG}$、$IM2_{OBJA}$に各々対応付けられたカラールックアップテーブル$CLT2_{BG}$，$CLT2_{OBJA}$を、ROM2からカラールックアップテーブル部65（図17参照）のオブジェクトA用テーブルエリアCLTAにDMA転送する。続いて、ステップSS12では、画像データ$IM2_{OBJB}$に対応付けられたカラールックアップテーブル$CLT2_{OBJB}$を、ROM2からカラールックアップテーブル部65のオブジェクトB用テーブルエリアCLTBにDMA転送する。そして、この後にステップSS13に進み、レジスタBGFに格納されるバンド画面転送フラグをゼロリセットして転送完了を表わし、ステップSS14に処理を進める。

【0127】ステップSS14に進むと、CPU1は、前述した画面切替処理ルーチンによりRAM3に格納されたバンド画面用OBJAコード$OBJ2_{OBJA}$を読み出してVDP5内部のオブジェクトメモリ（OM）52へ転送し、続くステップSS15において、同様にRAM3からバンド画面用OBJBコード$OBJ2_{OBJB}$を読み出してVDP5内部のオブジェクトメモリ（OM）52へ転送する。これにより、VDP5は次フレームで表示すべきバンド画面を発生する。なお、上述のステップSS8の判断結果が「NO」の場合、すなわち、既に転送が完了してバンド画面転送フラグがゼロリセットされている時には、上記ステップSS14，SS15を実行してバンド画面用OBJAコード$OBJ2_{OBJA}$とバンド画面用OBJBコード$OBJ2_{OBJB}$とをそれぞれオブジェクトメモリ（OM）52へ転送する。

【0128】③ジャンルセレクト画面モード

この場合、図44に示すステップSS16の判断結果が「YES」となり、ステップSS17に処理を進める。ステップSS17では、レジスタJSFに格納されるジャンルセレクト画面転送フラグが「1」であるか否かを判断する。このジャンルセレクト画面転送フラグは、ジャンルセレクト画面に切替えられた時点で「1」になるから、ここでの判断結果は「YES」となり、次のステップSS18に処理を進める。ステップSS18では、ジャンルセレクト画面を形成するバックグラウンド画像データ$IM3_{BG}$、オブジェクトA画像データ$IM3_{OBJA}$（図7参照）をそれぞれROM2からVRAM6へDMA転送する。続いて、ステップSS19では、ジャンルセレクト画面を形成するおよびオブジェクトB画像データ$IM3_{OBJB}$をROM2からVDP5内部のキャラジェネメモリ（CGM）54へDMA転送する。

【0129】次いで、ステップSS20に進むと、CPU1は上記画像データ$IM3_{BG}$、$IM3_{OBJA}$に各々対応付けられたカラールックアップテーブル$CLT3_{BG}$，$CLT3_{OBJA}$を、ROM2からカラールックアップテーブル部65内部のオブジェクトA用テーブルエリアCLTAにDMA転送する。続いて、ステップSS21では、画像データ$IM3_{OBJB}$に対応付けられたカラールックアップテーブル$CLT3_{OBJB}$を、ROM2からカラールックアップテーブル部65内部のオブジェクトB用テーブルエリアCLTBにDMA転送する。そして、この後にステップSS22に進み、レジスタJSFに格納されるバンド画面転送フラグをゼロリセットして転送完了を表わし、ステップSS23に処理を進める。

【0130】ステップSS23に進むと、CPU1は、前述した画面切替処理ルーチンによりRAM3に格納されたジャンルセレクト画面用OBJAコードOBJ3$_{OBJA}$を読み出してVDP5内部のオブジェクトメモリ（OM）52へ転送し、続くステップSS24において、同様にRAM3からジャンルセレクト画面用OBJBコードOBJ3$_{OBJB}$を読み出してVDP5内部のオブジェクトメモリ（OM）52へ転送する。これにより、VDP5は次フレームで表示すべきジャンルセレクト画面を発生する。なお、上述のステップSS17の判断結果が「NO」の場合、すなわち、既に転送が完了してジャンルセレクト画面転送フラグがゼロリセットされている時には、上記ステップSS23，SS24を実行してジャンルセレクト画面用OBJAコードOBJ3$_{OBJA}$、ジャンルセレクト画面用OBJBコードOBJ3$_{OBJB}$をオブジェクトメモリ（OM）52へ転送する。

【0131】④トータルコントロール画面モード

この場合、ステップSS25の判断結果が「YES」となり、ステップSS26に処理を進める。ステップSS26では、レジスタTCFに格納されるトータルコントロール画面転送フラグが「1」であるか否かを判断する。このフラグは、トータルコントロール画面に切替えられた時点で「1」になるから、ここでの判断結果は「YES」となり、次のステップSS27に処理を進める。ステップSS27では、トータルコントロール画面を形成するバックグラウンド画像データ$IM4_{BG}$、オブ

ジェクトA画像データＩＭ4_{OBJA}（図７参照）をそれぞれＲＯＭ２からＶＲＡＭ６へＤＭＡ転送する。

【０１３２】次いで、ステップＳＳ２８に進むと、ＣＰＵ１は上記画像データＩＭ4_{BG}、ＩＭ4_{OBJA}に各々対応付けられたカラールックアップテーブルＣＬＴ4_{BG}，ＣＬＴ4_{OBJA}を、ＲＯＭ２からカラールックアップテーブル部６５内部のオブジェクトＡ用テーブルエリアＣＬＴＡにＤＭＡ転送する。そして、この後にステップＳＳ２９に進み、レジスタＴＣＦに格納されるトータルコントロール画面転送フラグをゼロリセットして転送完了を表わし、ステップＳＳ３０に処理を進める。ステップＳＳ３０では、前述した画面切替処理ルーチンによりＲＡＭ３に格納されたトータルコントロール画面用ＯＢＪＡコードＯＢＪ4_{OBJA}を読み出してＶＤＰ５内部のオブジェクトメモリ（ＯＭ）５２へ転送する。これにより、ＶＤＰ５は次フレームで表示すべきトータルコントロール画面を発生する。なお、上述のステップＳＳ２５の判断結果が「ＮＯ」の場合、すなわち、既に転送が完了してトータルコントロール画面転送フラグがゼロリセットされている時には、上記ステップＳＳ３０を実行してトータルコントロール画面用ＯＢＪＡコードＯＢＪ4_{OBJA}をオブジェクトメモリ（ＯＭ）５２へ転送する。

【０１３３】⑤バンドセーブ画面モード

この場合、ステップＳＳ３１の判断結果が「ＹＥＳ」となり、ステップＳＳ３２に処理を進める。ステップＳＳ３２では、レジスタＢＳＦに格納されるバンドセーブ画面転送フラグが「１」であるか否かを判断する。この転送フラグは、バンドセーブ画面に切替えられた時点で「１」になるから、ここでの判断結果は「ＹＥＳ」となり、次のステップＳＳ３３に処理を進める。ステップＳＳ３３では、バンドセーブ画面を形成するバックグラウンド画像データＩＭ5_{BG}、オブジェクトＡ画像データＩＭ5_{OBJA}（図７参照）をそれぞれＲＯＭ２からＶＲＡＭ６へＤＭＡ転送する。続いて、ステップＳＳ３４では、バンドセーブ画面を形成するおよびオブジェクトＢ画像データＩＭ5_{OBJB}をＲＯＭ２からＶＤＰ５内部のキャラジェネメモリ（ＣＧＭ）５４へＤＭＡ転送する。

【０１３４】次いで、ステップＳＳ３５に進むと、ＣＰＵ１は上記画像データＩＭ5_{BG}、ＩＭ5_{OBJA}に各々対応付けられたカラールックアップテーブルＣＬＴ5_{BG}，ＣＬＴ5_{OBJA}を、ＲＯＭ２からカラールックアップテーブル部６５内部のオブジェクトＡ用テーブルエリアＣＬＴＡにＤＭＡ転送する。続いて、ステップＳＳ３６では、画像データＩＭ5_{OBJB}に対応付けられたカラールックアップテーブルＣＬＴ5_{OBJB}を、ＲＯＭ２からカラールックアップテーブル部６５内部のオブジェクトＢ用テーブルエリアＣＬＴＢにＤＭＡ転送する。そして、この後にステップＳＳ３７に進み、レジスタＢＳＦに格納されるバンドセーブ画面転送フラグをゼロリセットして転送完了を表わし、ステップＳＳ３８に処理を進める。

【０１３５】ステップＳＳ３８に進むと、ＣＰＵ１は、前述した画面切替処理ルーチンによりＲＡＭ３に格納されたバンドセーブ画面用ＯＢＪＡコードＯＢＪ6_{OBJA}を読み出してＶＤＰ５内部のオブジェクトメモリ（ＯＭ）５２へ転送し、続くステップＳＳ３９において、同様にＲＡＭ３からバンドセーブ画面用ＯＢＪＢコードＯＢＪ6_{OBJB}を読み出してＶＤＰ５内部のオブジェクトメモリ（ＯＭ）５２へ転送する。これにより、ＶＤＰ５は次フレームで表示すべきバンドセーブ画面を発生する。なお、上述のステップＳＳ３２の判断結果が「ＮＯ」の場合、すなわち、既に転送が完了してバンドセーブ画面転送フラグがゼロリセットされている時には、上記ステップＳＳ３８，ＳＳ３９を実行してバンドセーブ画面用ＯＢＪＡコードＯＢＪ6_{OBJA}、バンドセーブ画面用ＯＢＪＢコードＯＢＪ6_{OBJB}をオブジェクトメモリ（ＯＭ）５２へ転送する。

【０１３６】以上説明したように、本実施例では、画面タイトルなどの画面上で固定的に表示される文字をオブジェクトＡ画像で形成する一方、このオブジェクトＡ画像上で表示変更される文字を表示する表示領域（表示行）に、予め表示可能な最大文字数分のオブジェクトＢ画像を配置しておき、その文字を書き換える時にはその表示位置に対応するＯＢＪＢコード中のキャラクタネーム（文字種類）だけを変更し、文字を消去する際にはそのＯＢＪＢコード中のカラーブロックを透明指定させて非表示状態にしている。したがって、従来の画像制御装置のように、画面上で表示変更される文字と画面タイトルなどの固定的に表示される文字との両者について、その種類と表示座標とを個々に管理したり、文字消去の際には対応するオブジェクトコード中の表示位置を画面表示領域外へ移動させる処理が不要になる。このため、高速なＣＰＵを用いることなく処理速度の向上を図ることが可能となり、これを換言すれば、コストアップを招致することなく処理速度の向上が可能になっている。

【０１３７】また、上述した実施例にあっては、マウスなどのポインティングデバイスを備えずとも、十字キー１０９やエンター１１１の操作に応じて表示色を変更すべきオブジェクトＡ画像を指定し、これに対応するＯＢＪＡコード中のカラーブロック番号を、「ノーマル色」、「セレクト色」および「アクティブ色」のいずれかを示すものに書き換えるようにしたので、キー操作内容に対応して画像表示態様を可変とするＧＵＩ環境が実現でき、この結果、誤操作を防ぐ等、操作性向上が図られている。これと類似する点として、上述した実施例では、画面表示される「ボタン」を、オブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃ（図２４（ロ）参照）で構成し、画像変更する操作形態の内容に応じてこれらオブジェクトＯＢＪＡ－Ａ～ＯＢＪＡ－Ｃの表示色を変化させ、該当「ボタン」の状態（ノーマル状態、セレクト状態およびアクティブ状態）を表示するため、視認性に優れたＧ

ＵＩ環境を達成している。

【０１３８】更に、上述した実施例では、画面上で固定的に表示されるオブジェクトＡ画像のＯＢＪＡコードに反転フラグＸを設け、このフラグＸの値に応じてオブジェクトＡ画像データをラインバッファへ格納する際の書込み順序をアドレス順するか、あるいはそれと逆順にするかを指定できるようにした為、１つの画像データをアドレス順に書込んだ通常の画像として表示させたり、これとは逆順に書込んで像の左右が反転した反転画像を表示することが可能になる。つまり、１つの画像データを通常に表示するものと、反転表示するものとの２画像を形成し得るようになるから、表示し得る画像の種類が増し、メモリ容量の少ないシステムにとっては画像データを効率的に使用することができる。

【０１３９】なお、この実施例においては、本発明による画像制御装置を、ディスプレイに接続される楽音制御装置に適用した場合について言及したが、本発明の要旨は言うまでもなくこれに限定されず、例えば、パーソナルコンピュータやワードプロセッサ等の情報機器はもとより携帯端末装置（ページャー、ＰＨＳ等）やビデオゲーム装置にも適用可能である。要は、画像を用いたユーザー・インタフェースを実現する装置であれば、本発明の要旨を適用でき、そうすることで視認性に優れ、誤操作を防ぐ操作環境を実現しながら、製品のコストアップを招致せずに表示制御の処理速度を向上せしめる、という顕著な効果が得られる。

【０１４０】

【発明の効果】本発明によれば、表示態様を変更可能とした変更画像の表示領域を予め画面上で定め、画像配置手段がこの表示領域内に表示し得る数分の変更画像を配置しておくと、変更手段によって表示領域内に配置される変更画像のいずれかを指定してその表示態様を変更すると、変更された表示態様に応じて変更画像が更新され、これが表示制御手段によって前記表示領域内での配置に従って画面表示される。つまり、表示態様が変更される画像だけが更新対象となるから、高速なＣＰＵを用いることなく処理速度の向上を図ることが可能となり、換言すれば、コストアップを招致することなく処理速度を向上することができる。

【図面の簡単な説明】

【図１】本発明による一実施例の外観を示す外観図である。

【図２】同実施例の全体構成を示すブロック図である。

【図３】同実施例におけるＲＯＭ２に格納されるオブジェクトコード種類を説明するためのメモリマップである。

【図４】同実施例におけるＲＯＭ２に格納されるオブジェクトコード種類を説明するためのメモリマップである。

【図５】ＲＯＭ２に格納されるＯＢＪＡコードの構成を示す図である。

【図６】ＲＯＭ２に格納されるＯＢＪＢコードの構成を示す図である。

【図７】同実施例におけるＲＯＭ２に格納される画像データ種類を説明するためのメモリマップである。

【図８】ＲＯＭ２に格納されるオブジェクトＢ画像データの構成を示す図である。

【図９】ＲＯＭ２に格納されるオブジェクトＢ画像データの構成を示す図である。

【図１０】ＲＯＭ２に格納されるオブジェクトＡ画像データの構成を示す図である。

【図１１】ＲＯＭ２に格納されるオブジェクトＡ画像データの構成を示す図である。

【図１２】同実施例におけるＲＯＭ２に格納されるカラールックアップテーブル種類を説明するためのメモリマップである。

【図１３】バンド画面に用いられるオブジェクトＯＢＪＡ用カラールックアップテーブルＣＬＴ2_{OBJA}の構成を示す図である。

【図１４】ジャンルセレクト画面に用いられるオブジェクトＯＢＪＡ用カラールックアップテーブルＣＬＴ3_{OBJA}の構成を示す図である。

【図１５】ジャンルセレクト画面に用いられるオブジェクトＯＢＪＢ用カラールックアップテーブルＣＬＴ3_{OBJB}の構成を示す図である。

【図１６】トータルコントロール画面に用いられるオブジェクトＯＢＪＡ用カラールックアップテーブルＣＬＴ4_{OBJA}の構成を示す図である。

【図１７】同実施例におけるＶＤＰ５の構成を示すブロック図である。

【図１８】バンド画面の構成例を示す図である。

【図１９】バンドメンバー変更画面の構成例を示す図である。

【図２０】ジャンルセレクト画面の構成例を示す図である。

【図２１】ジャンルセレクト画面に表示されるウインドウＷＩＮを構成するオブジェクト種類を説明するための図である。

【図２２】バンドセーブ画面の構成例を示す図である。

【図２３】トータルコントロール画面の構成例を示す図である。

【図２４】トータルコントロール画面に表示されるボタンを構成するオブジェクトの表示形態およびオブジェクト種類を示す図である。

【図２５】本実施例の概略動作を示すフローチャートである。

【図２６】上記概略動作におけるメインフローを説明するためのフローチャートである。

【図２７】メインフローにおける画像処理の内容を説明するためのフローチャートである。

【図２８】画像処理における画面切替処理ルーチンの動作を説明するためのフローチャートである。

【図２９】画像処理におけるバンド画面処理ルーチンの動作を説明するためのフローチャートである。

【図３０】画像処理における文字列変更処理ルーチンの動作を説明するためのフローチャートである。

【図３１】画像処理におけるカラーブロック変更処理（ＯＢＪＢ）ルーチンの動作を説明するためのフローチャートである。

【図３２】画像処理におけるバンドメンバー変更処理ルーチンの動作を説明するためのフローチャートである。

【図３３】画像処理におけるオブジェクト移動処理ルーチンの動作を説明するためのフローチャートである。

【図３４】画像処理におけるオブジェクト移動処理ルーチンの動作を説明するためのフローチャートである。

【図３５】画像処理におけるオブジェクト色変更処理ルーチンの動作を説明するためのフローチャートである。

【図３６】画像処理におけるオブジェクト色変更処理ルーチンの動作を説明するためのフローチャートである。

【図３７】画像処理におけるジャンルセレクト画面処理ルーチンの動作を説明するためのフローチャートである。

【図３８】画像処理におけるトータルコントロール画面処理ルーチンの動作を説明するためのフローチャートである。

【図３９】画像処理におけるボタン用カラーブロック変更処理ルーチンの動作を説明するためのフローチャート

である。

【図４０】画像処理におけるＢｅｅｐＯｎボタンカラーブロック変更処理ルーチンの動作を説明するためのフローチャートである。

【図４１】画像処理におけるバンドセーブ画面処理ルーチンの動作を説明するためのフローチャートである。

【図４２】画像処理におけるカラーブロック変更処理（ＯＢＪＡ）ルーチンの動作を説明するためのフローチャートである。

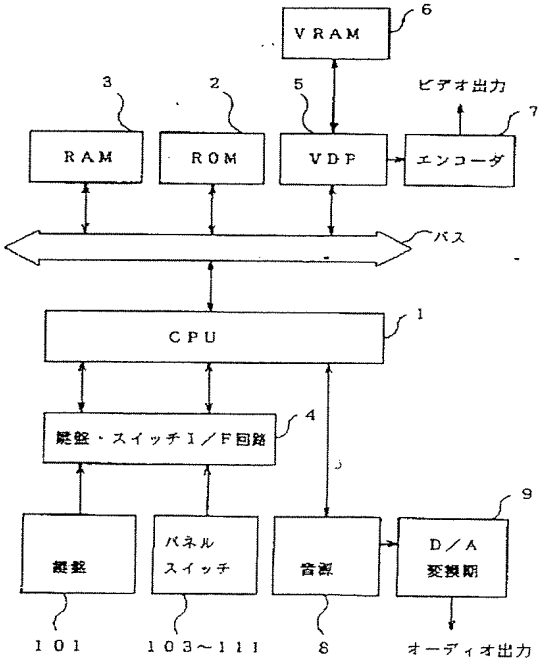
【図４３】画像処理におけるＶブランクインタラプト処理ルーチンの動作を説明するためのフローチャートである。

【図４４】画像処理におけるＶブランクインタラプト処理ルーチンの動作を説明するためのフローチャートである。

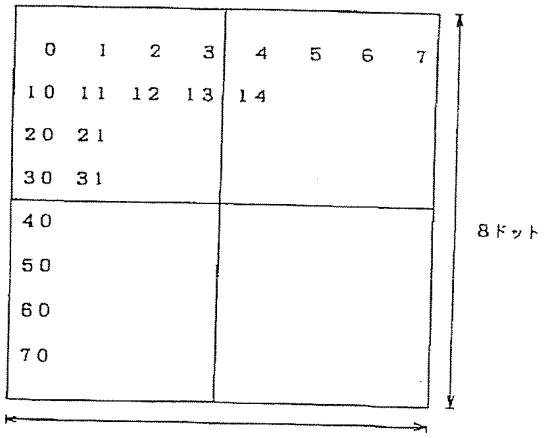
【符号の説明】

１　ＣＰＵ（画像配置手段、変更手段、表示制御手段）
２　ＲＯＭ（画像配置手段、表示制御手段）
３　ＲＡＭ（表示制御手段）
４　鍵盤・スイッチＩ／Ｆ回路
５　ＶＤＰ
６　ＶＲＡＭ
７　エンコーダ
８　音源
９　Ｄ／Ａ変換器
１０１　鍵盤
１０３～１１１　パネルスイッチ（変更手段）

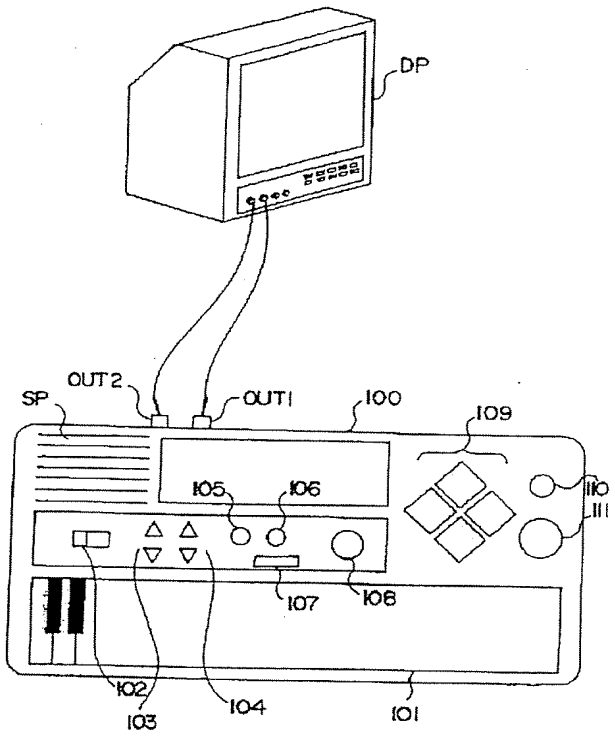
【図２】



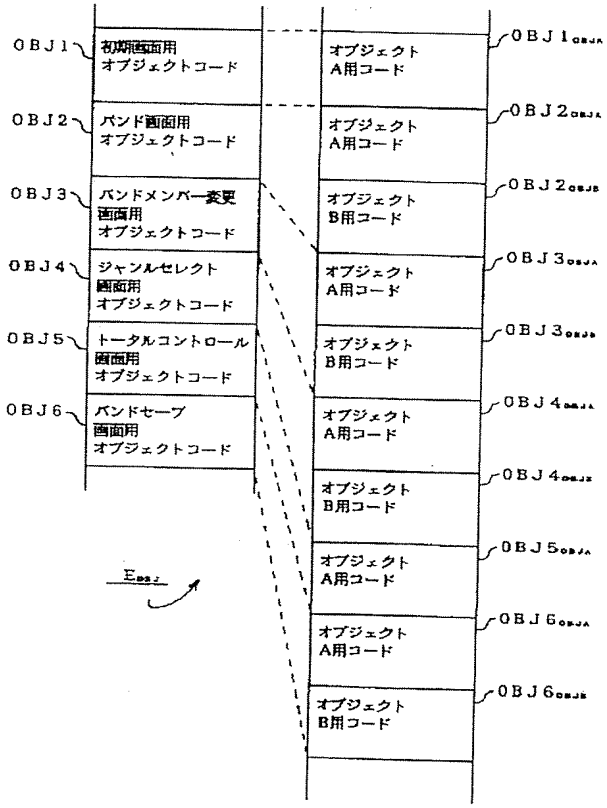
【図８】



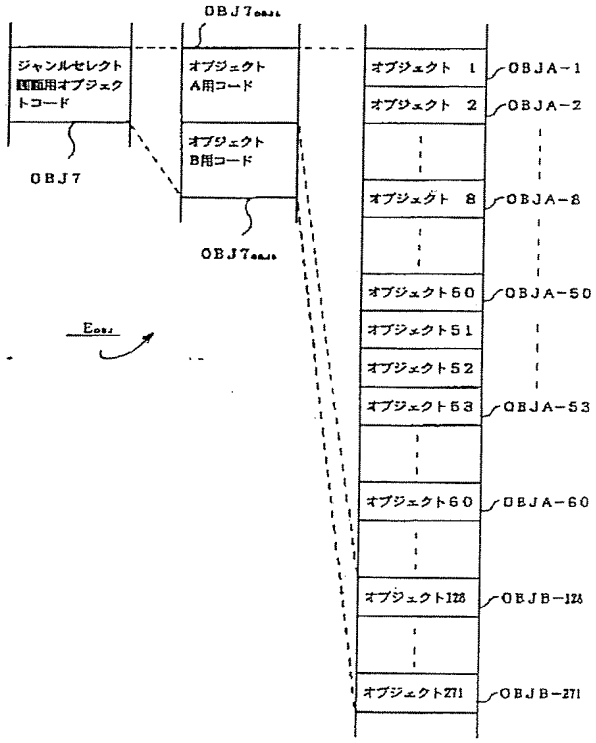
【図1】


DP
OUT2
OUT1
SP
100
109
105
106
110
111
107
108
102
103
104
101

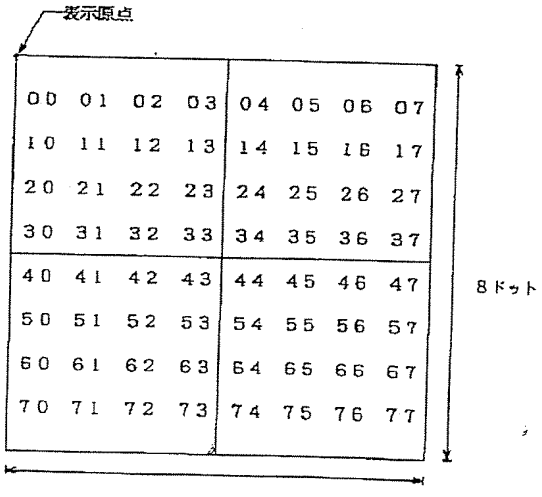
【図3】


OBJ1
初期画面用
オブジェクトコード
OBJ2
バンド画面用
オブジェクトコード
OBJ3
バンドメンバー変更
画面用
オブジェクトコード
OBJ4
ジャンルセレクト
画面用
オブジェクトコード
OBJ5
トータルコントロール
画面用
オブジェクトコード
OBJ6
バンドセーブ
画面用
オブジェクトコード
オブジェクト
A用コード
オブジェクト
B用コード
Exec

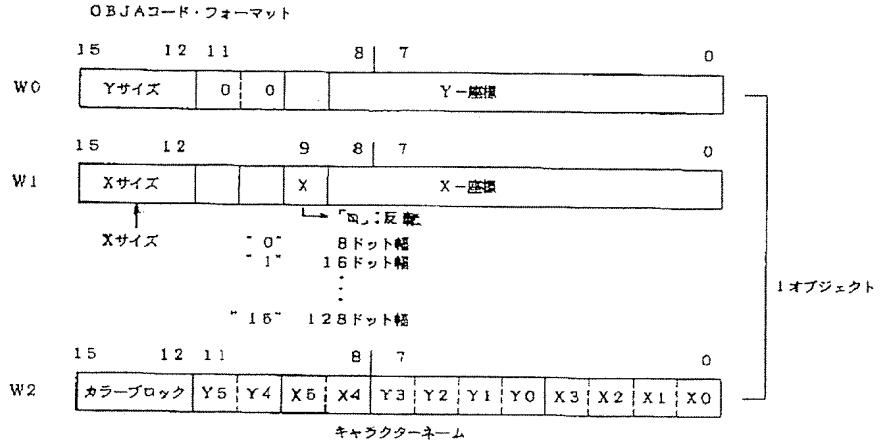
【図4】


OBJ7
ジャンルセレクト
画面用オブジェク
トコード
オブジェクト
A用コード
オブジェクト
B用コード
Exec
オブジェクト 1
OBJA-1
オブジェクト 2
OBJA-2
オブジェクト 8
OBJA-8
オブジェクト50
OBJA-50
オブジェクト51
オブジェクト52
オブジェクト53
OBJA-53
オブジェクト60
OBJA-60
オブジェクト128
OBJB-128
オブジェクト271
OBJB-271

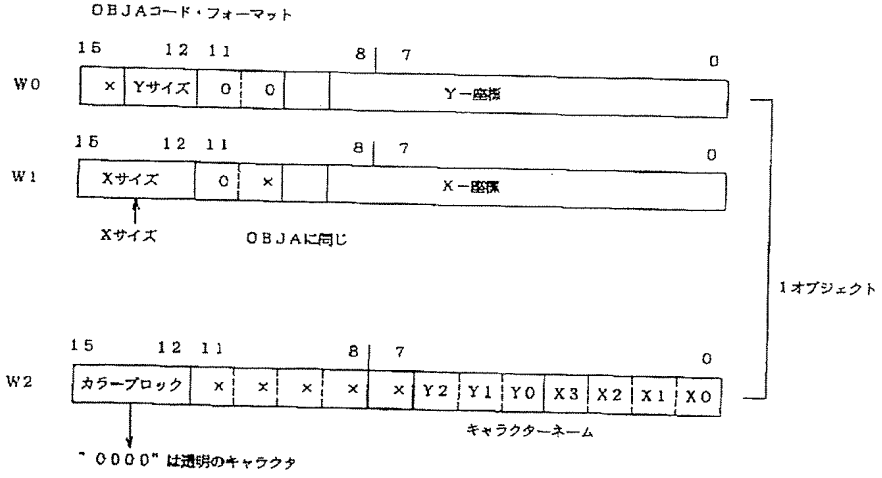
【図10】


表示原点
00 01 02 03 04 05 06 07
10 11 12 13 14 15 16 17
20 21 22 23 24 25 26 27
30 31 32 33 34 35 36 37
40 41 42 43 44 45 46 47
50 51 52 53 54 55 56 57
60 61 62 63 64 65 66 67
70 71 72 73 74 75 76 77
8ドット
8ドット

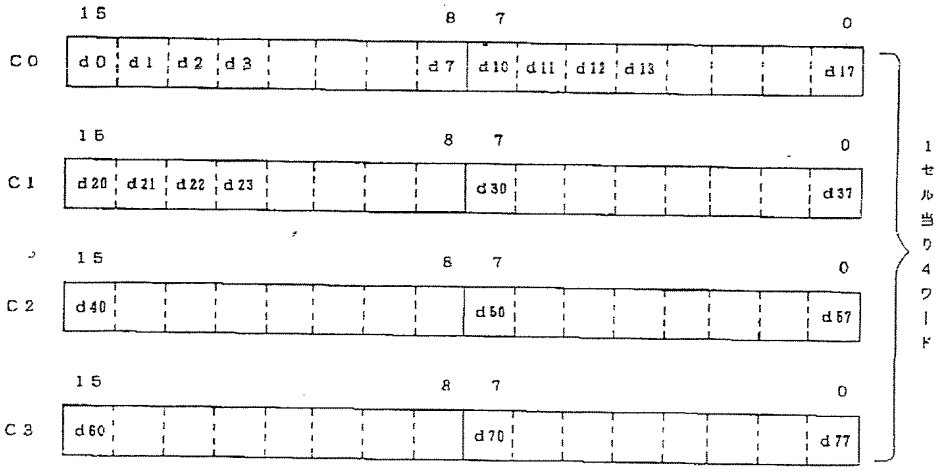
【図５】



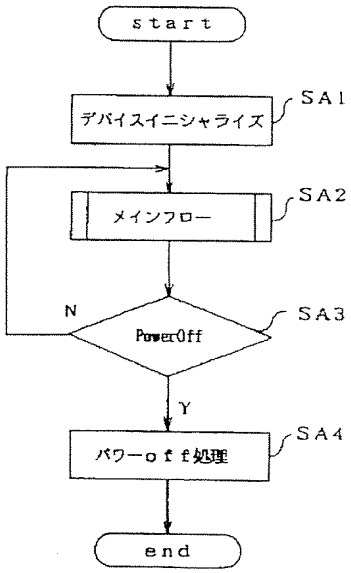
【図６】



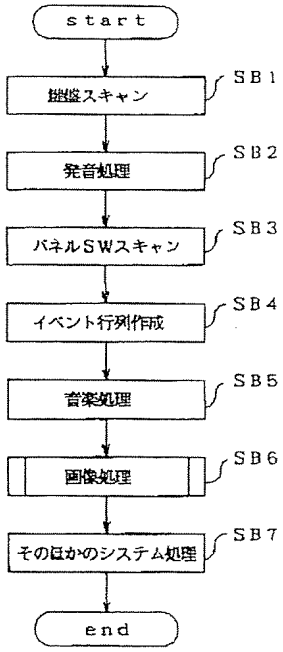
【図９】



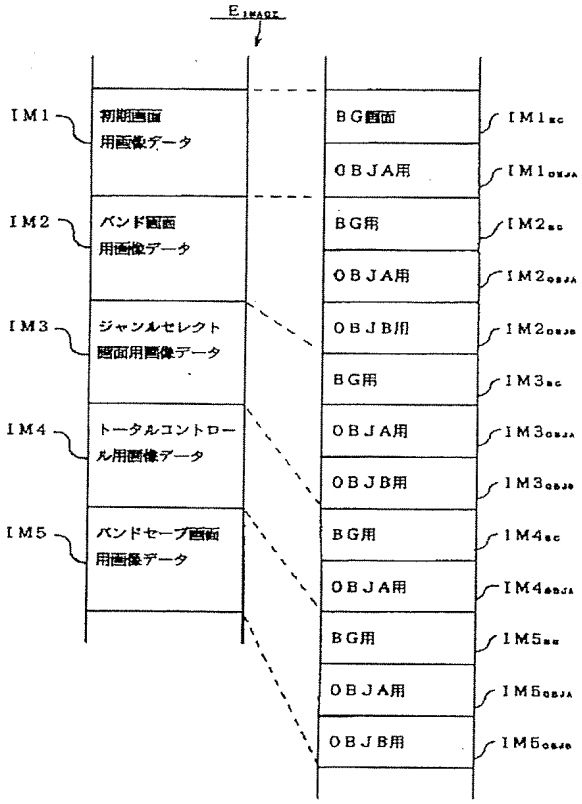
【図２５】



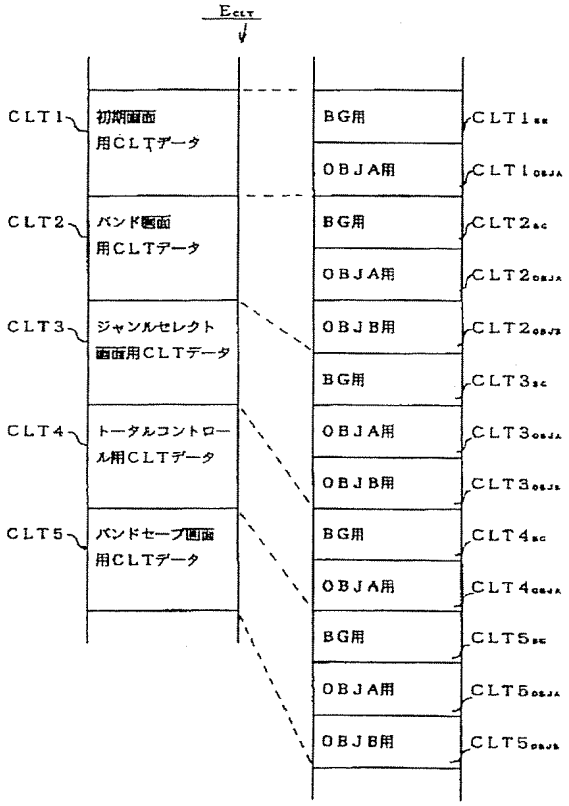
【図２６】



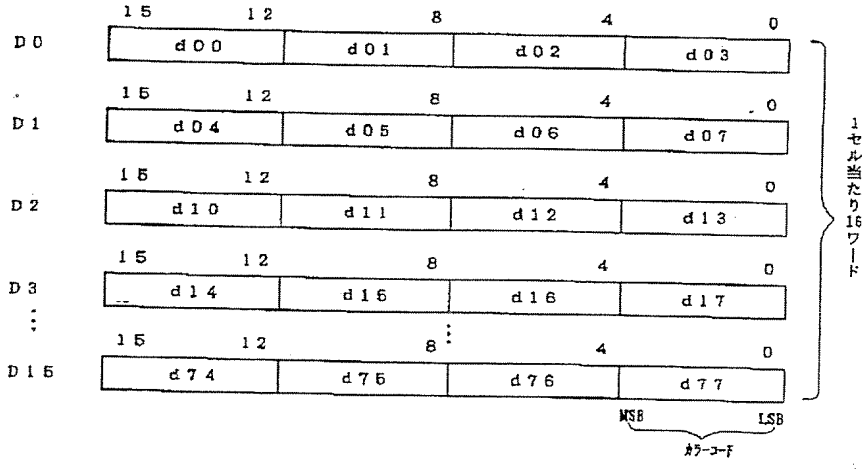
【図７】



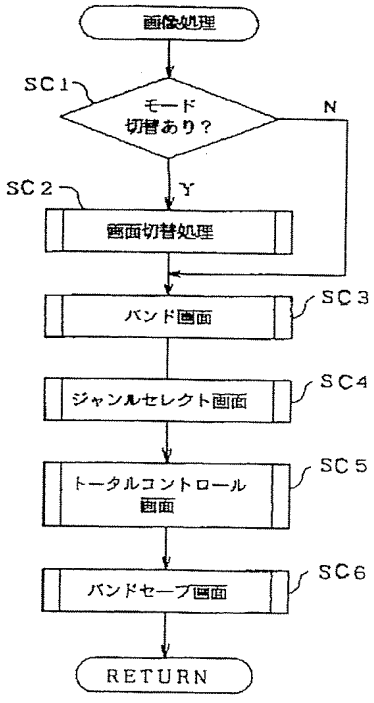
【図１２】



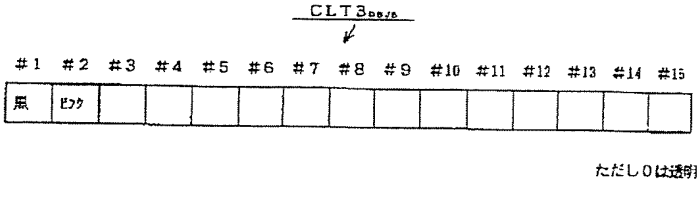
【図１１】



【図２７】

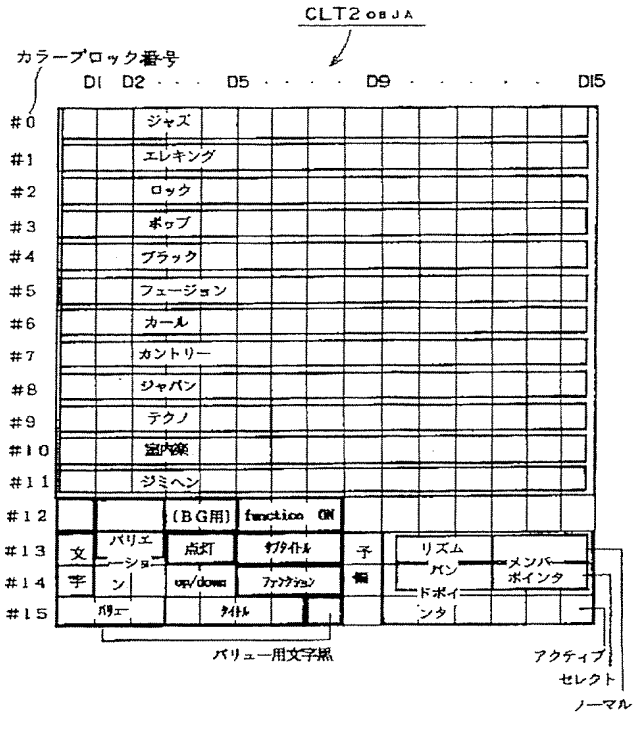


【図１５】

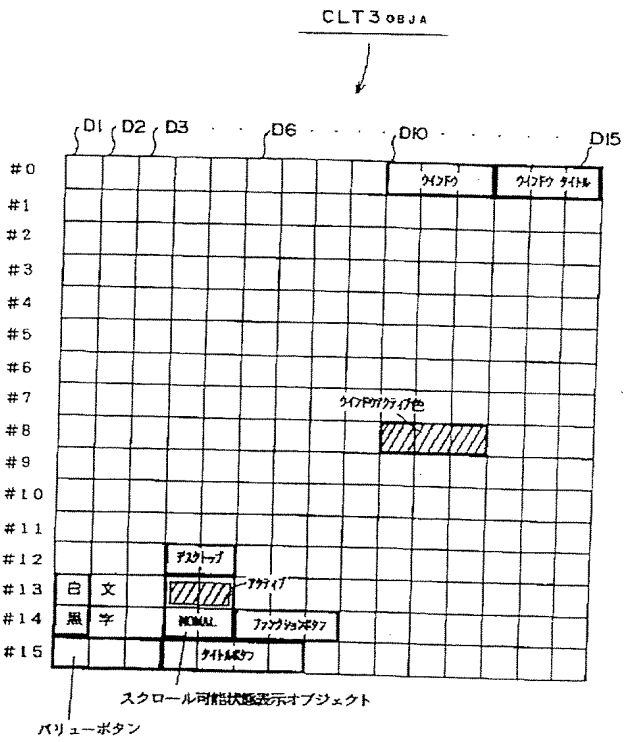


ただし0は透明

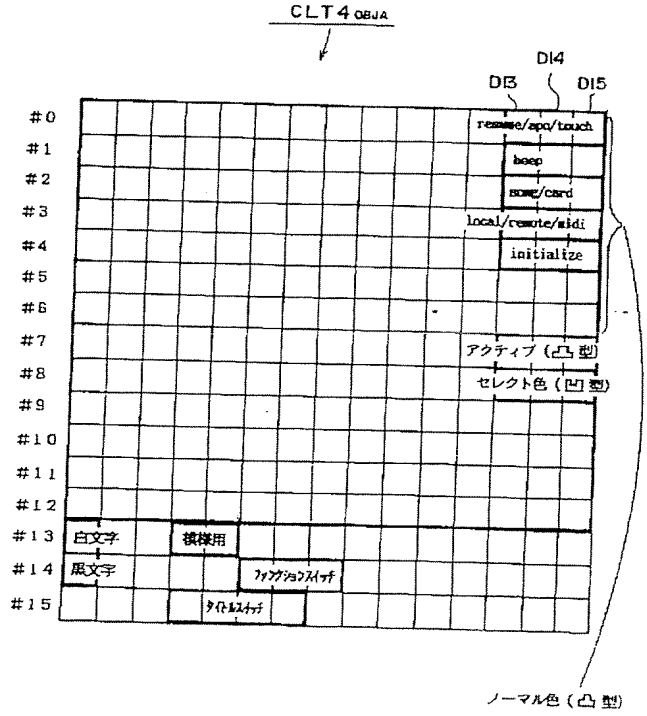
【図１３】



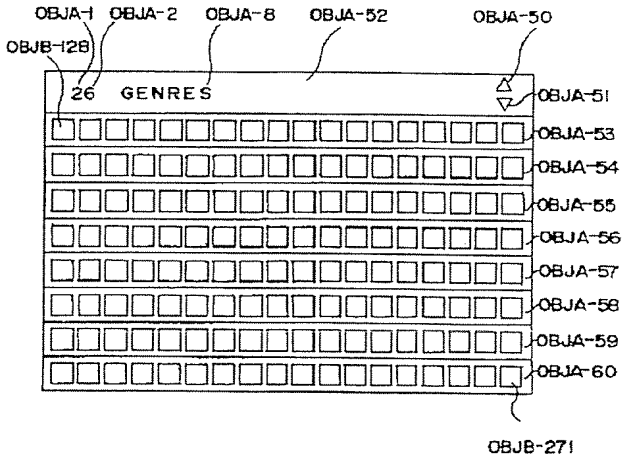
【図１４】



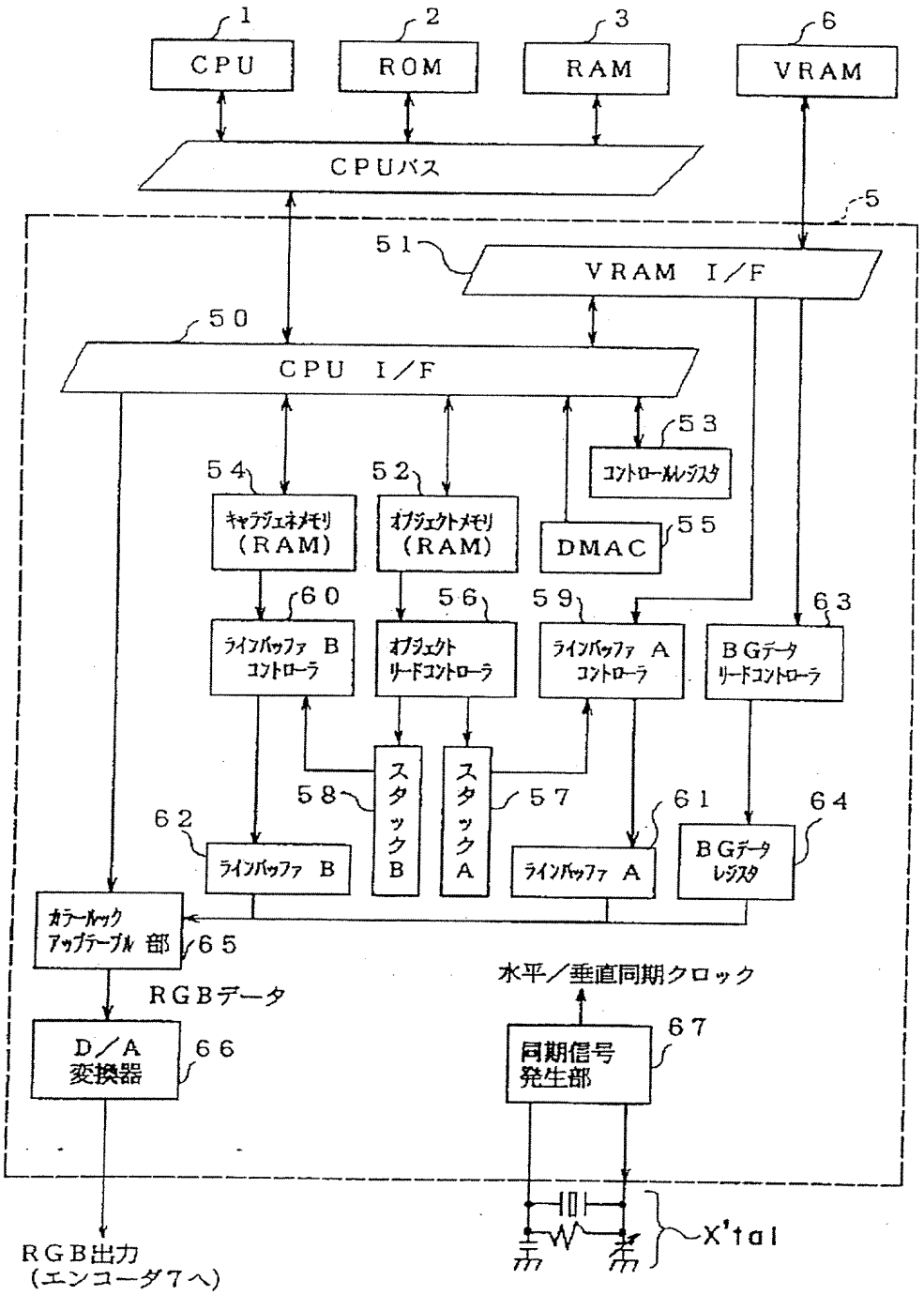
【図１６】



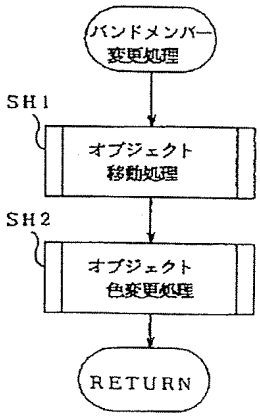
【図２１】



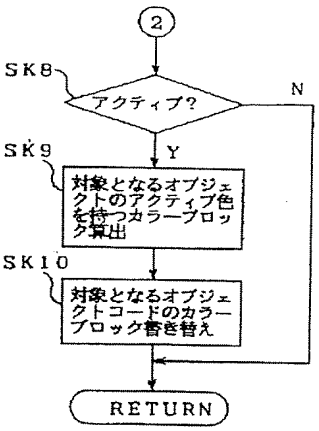
【図１７】


1
CPU
2
ROM
3
RAM
6
VRAM
CPUバス
5
51
VRAM I／F
50
CPU I／F
53
コントロールレジスタ
54
キャラジェネメモリ
(RAM)
52
オブジェクトメモリ
(RAM)
DMAC
55
60
ラインバッファ B
コントローラ
56
オブジェクト
リードコントローラ
59
ラインバッファ A
コントローラ
63
BGデータ
リードコントローラ
58
スタック
B
57
スタック
A
62
ラインバッファ B
61
ラインバッファ A
64
BGデータ
レジスタ
カラールック
アップテーブル 部
65
RGBデータ
D／A
変換器
66
水平／垂直同期クロック
同期信号
発生部
67
X'tal
RGB出力
(エンコーダ７へ)

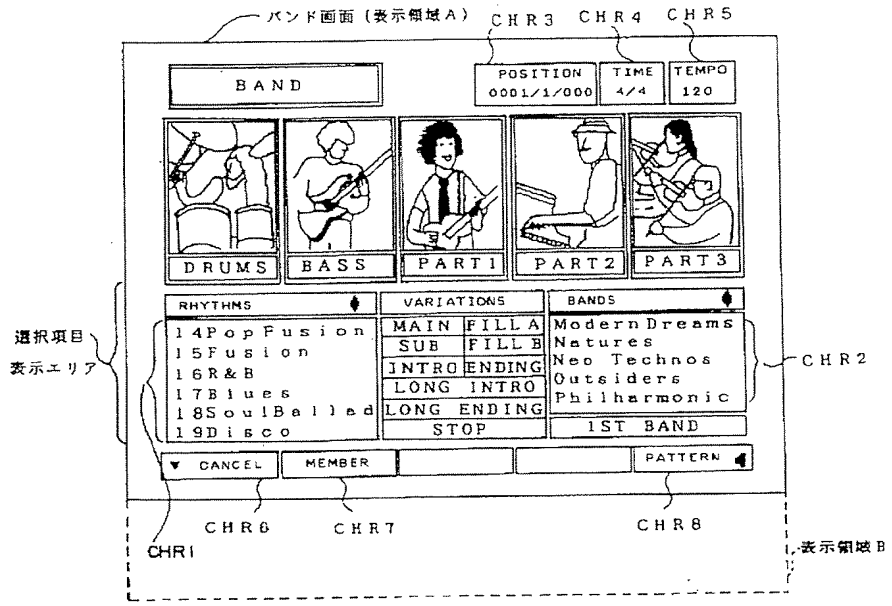
【図３２】


バンドメンバー
変更処理
SH1
オブジェクト
移動処理
SH2
オブジェクト
色変更処理
RETURN

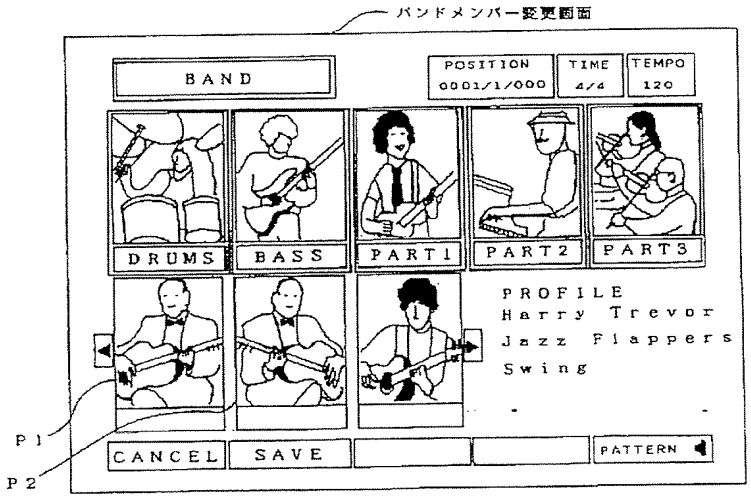
【図３６】


2
SK8
アクティブ?
N
Y
SK9
対象となるオブジェクトのアクティブ色を持つカラーブロック算出
SK10
対象となるオブジェクトコードのカラーブロック書き替え
RETURN

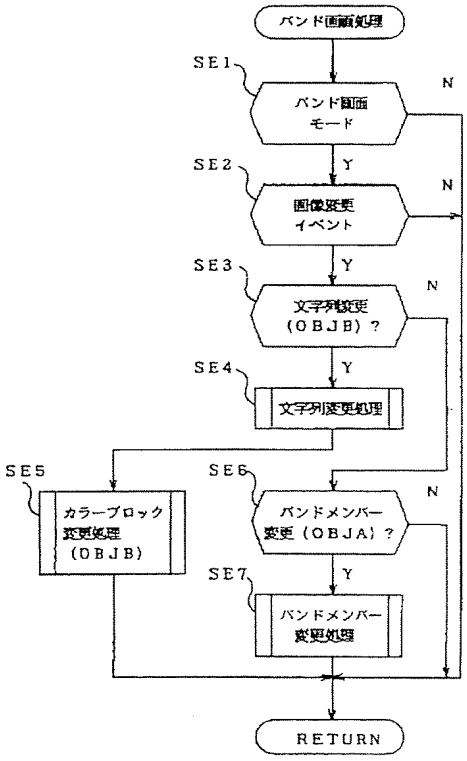
【図１８】


バンド画面（表示領域Ａ）
CHR3
CHR4
CHR5
BAND
POSITION 0001/1/000
TIME 4/4
TEMPO 120
DRUMS
BASS
PART1
PART2
PART3
RHYTHMS
VARIATIONS
BANDS
14PopFusion
15Fusion
16R&B
17Blues
18SoulBallad
19Disco
MAIN FILL A
SUB FILL B
INTRO ENDING
LONG INTRO
LONG ENDING
STOP
ModernDreams
Natures
Neo Technos
Outsiders
Philharmonic
1ST BAND
CANCEL
MEMBER
PATTERN
選択項目表示エリア
CHR2
CHR6
CHR7
CHR8
CHR1
表示領域Ｂ

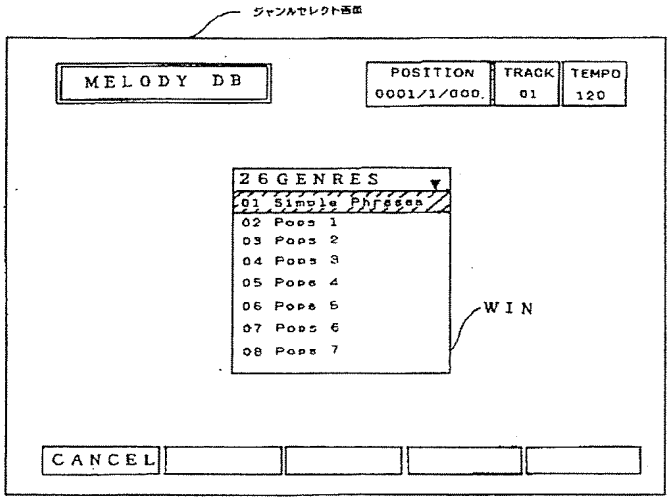
【図１９】


バンドメンバー変更画面
BAND
POSITION 0001/1/000
TIME 4/4
TEMPO 120
DRUMS
BASS
PART1
PART2
PART3
PROFILE
Harry Trevor
Jazz Flappers
Swing
P1
P2
CANCEL
SAVE
PATTERN

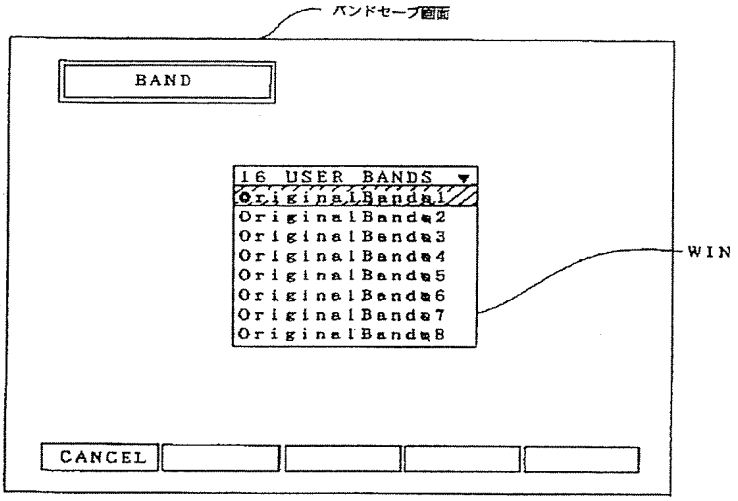
【図２９】


バンド画面処理
SE1
バンド画面モード
N
Y
SE2
画像変更イベント
N
Y
SE3
文字列変更(OBJB)？
N
Y
SE4
文字列変更処理
SE5
カラーブロック変更処理(OBJB)
SE6
バンドメンバー変更(OBJA)？
N
Y
SE7
バンドメンバー変更処理
RETURN

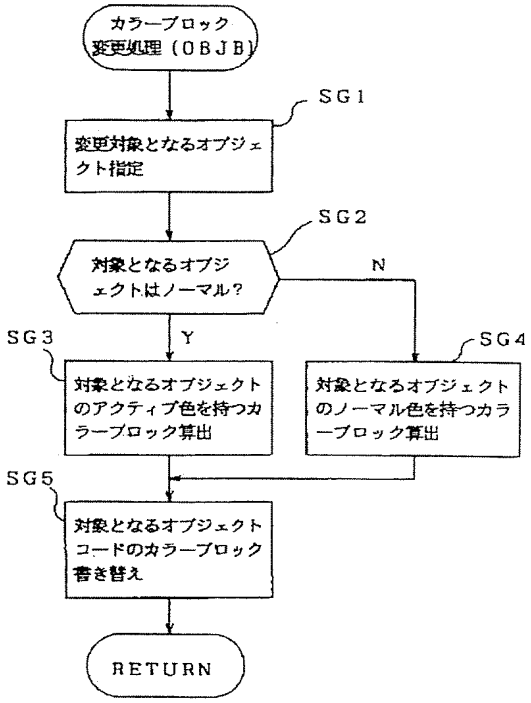
【図２０】



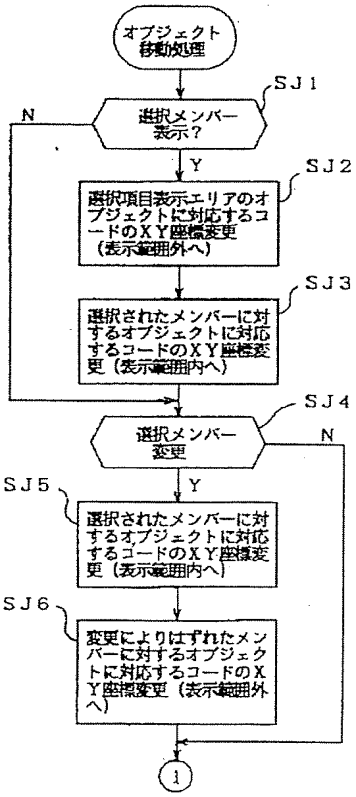
【図２２】



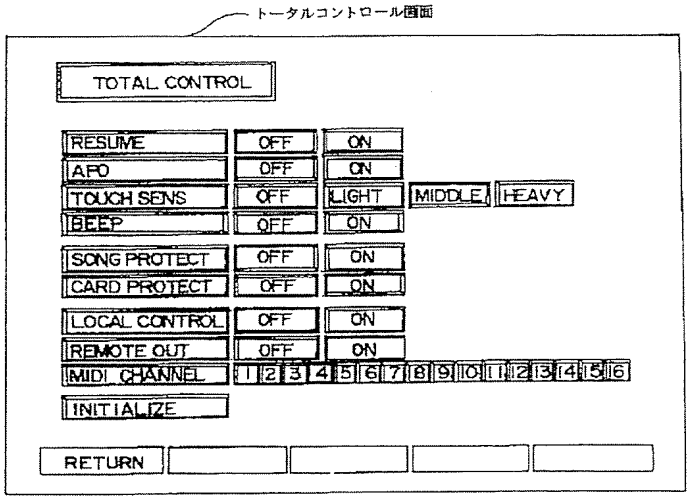
【図３１】



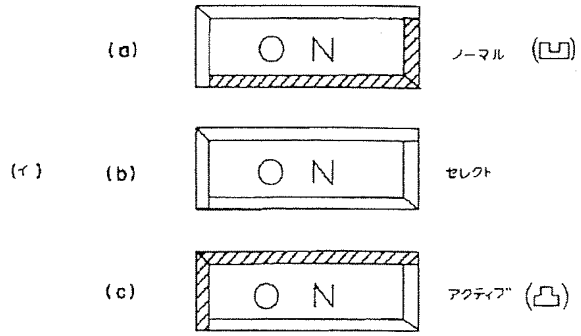
【図３３】



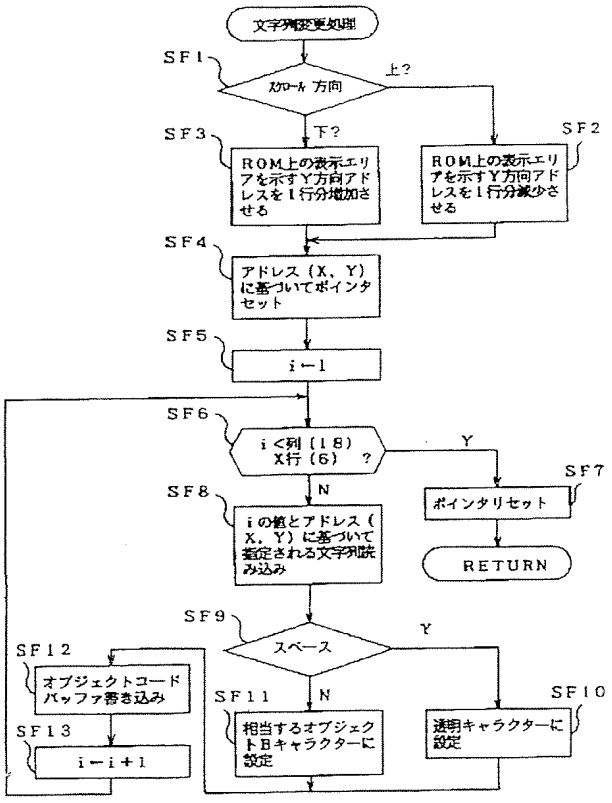
【図２３】

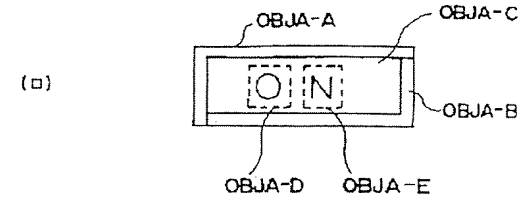


【図２４】

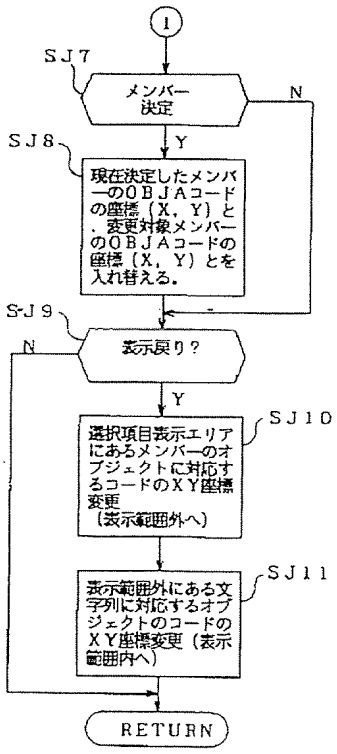


【図３０】

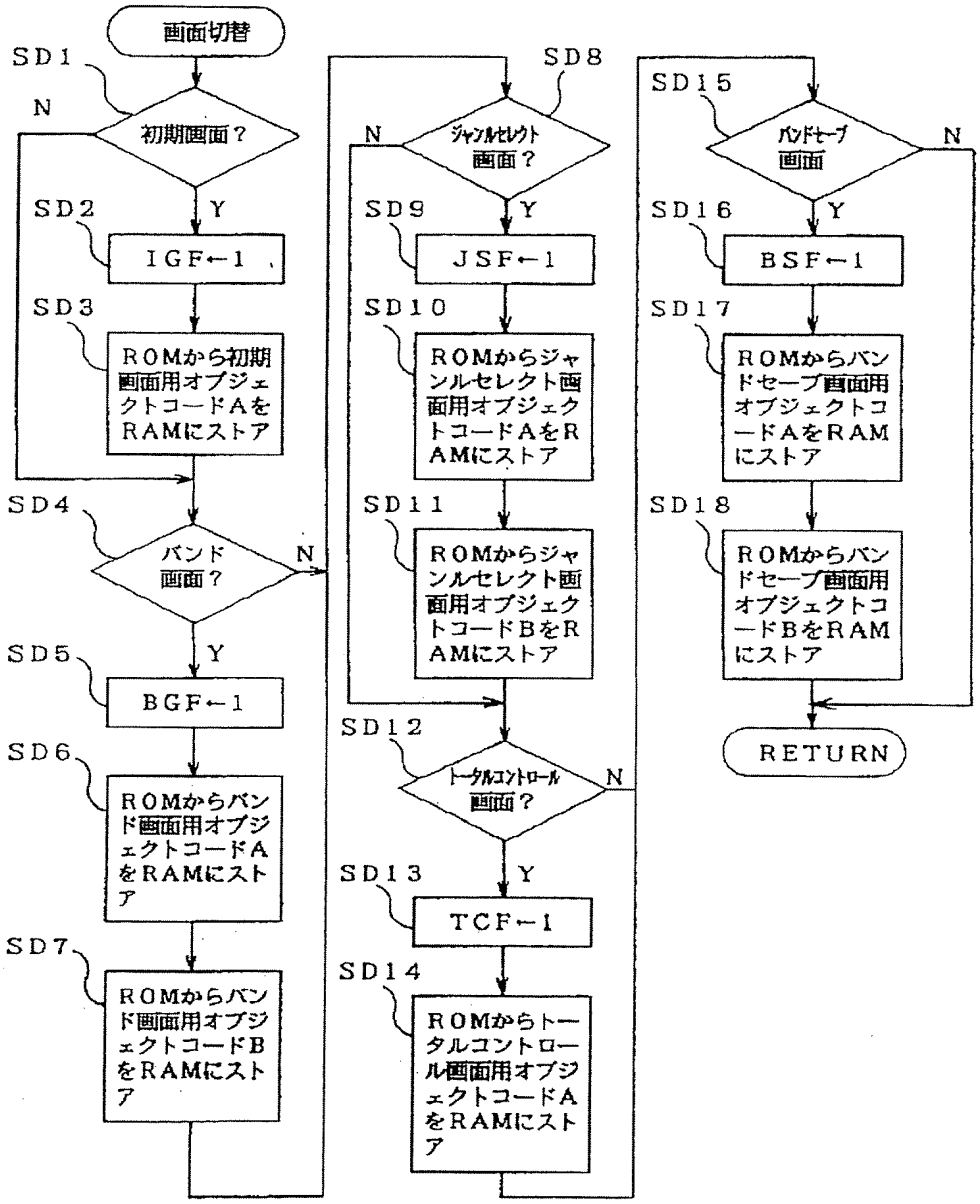




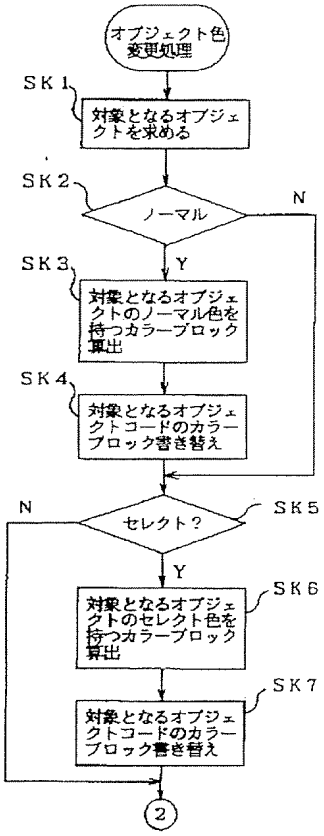
【図３４】



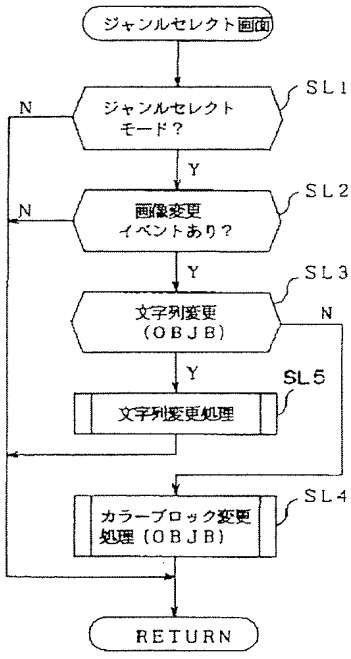
【図２８】


画面切替
SD1
初期画面？
N
Y
SD2
IGF←1
SD3
ROMから初期画面用オブジェクトコードAをRAMにストア
SD4
バンド画面？
N
Y
SD5
BGF←1
SD6
ROMからバンド画面用オブジェクトコードAをRAMにストア
SD7
ROMからバンド画面用オブジェクトコードBをRAMにストア
SD8
ジャンルセレクト画面？
N
Y
SD9
JSF←1
SD10
ROMからジャンルセレクト画面用オブジェクトコードAをRAMにストア
SD11
ROMからジャンルセレクト画面用オブジェクトコードBをRAMにストア
SD12
トータルコントロール画面？
N
Y
SD13
TCF←1
SD14
ROMからトータルコントロール画面用オブジェクトコードAをRAMにストア
SD15
バンドセーブ画面
N
Y
SD16
BSF←1
SD17
ROMからバンドセーブ画面用オブジェクトコードAをRAMにストア
SD18
ROMからバンドセーブ画面用オブジェクトコードBをRAMにストア
RETURN

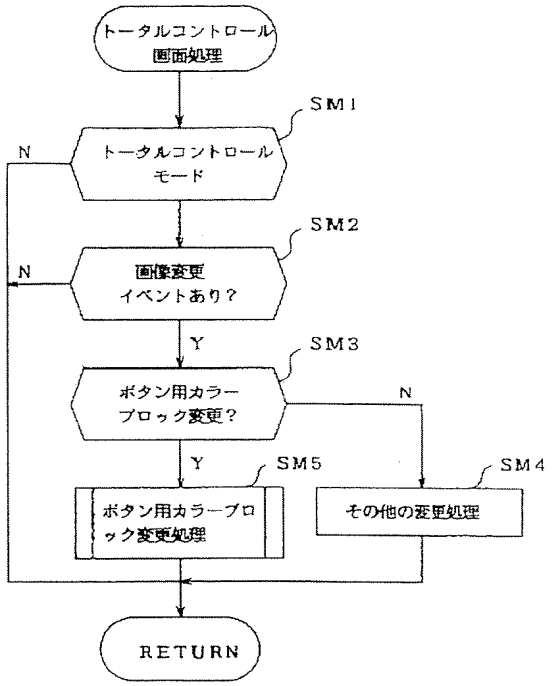
【図３５】



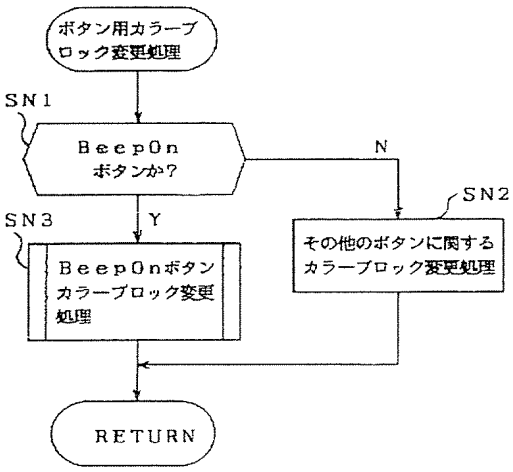
【図３７】



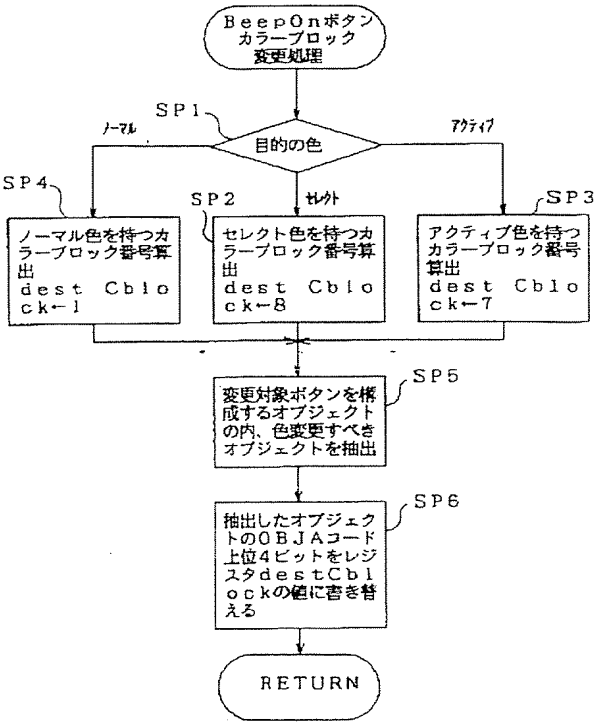
【図３８】



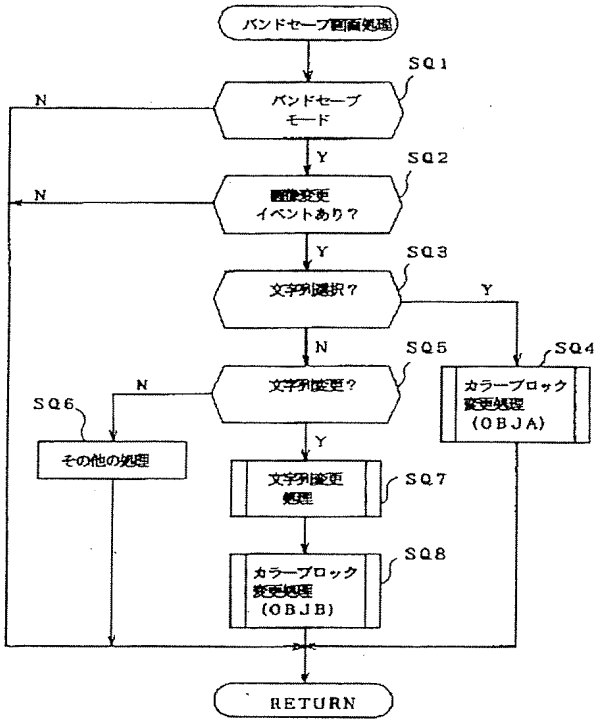
【図３９】



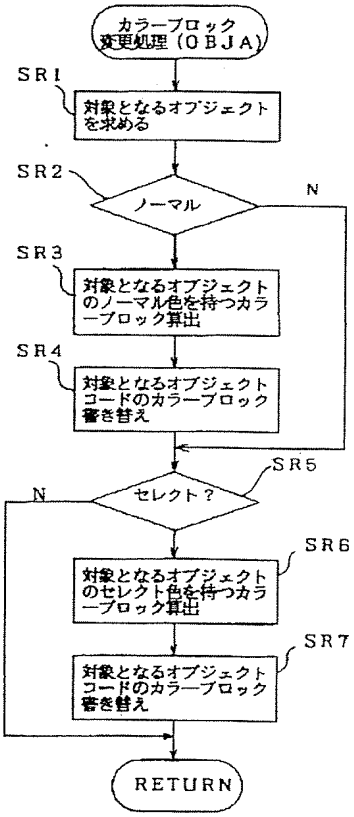
【図４０】



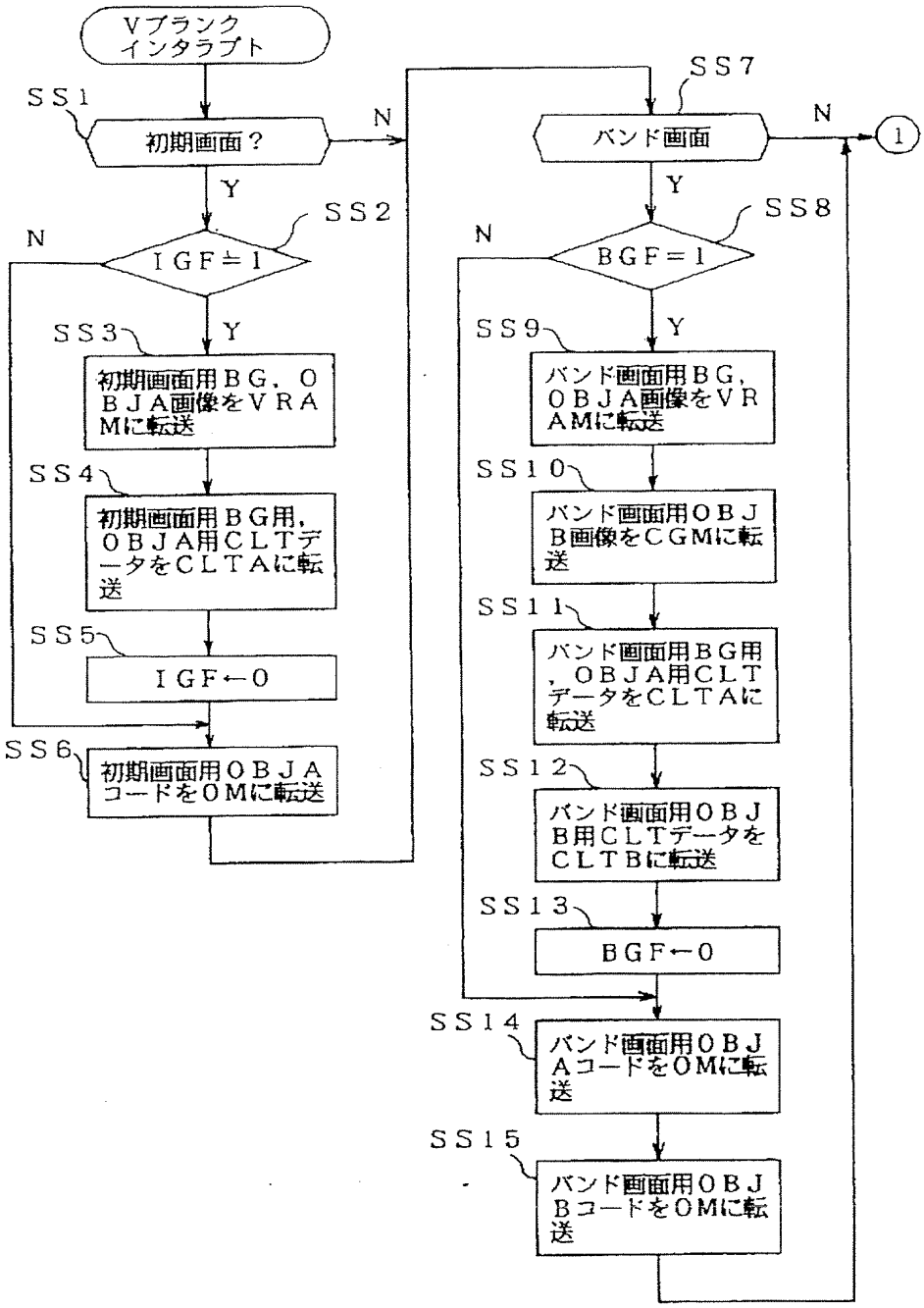
【図４１】


バンドセーブ画面処理
SQ1
バンドセーブ
モード
N
Y
SQ2
画像変更
イベントあり？
N
Y
SQ3
文字列選択？
Y
N
SQ5
SQ4
文字列変更？
カラーブロック
変更処理
(OBJA)
N
SQ6
その他の処理
Y
文字列変更
処理
SQ7
SQ8
カラーブロック
変更処理
(OBJB)
RETURN

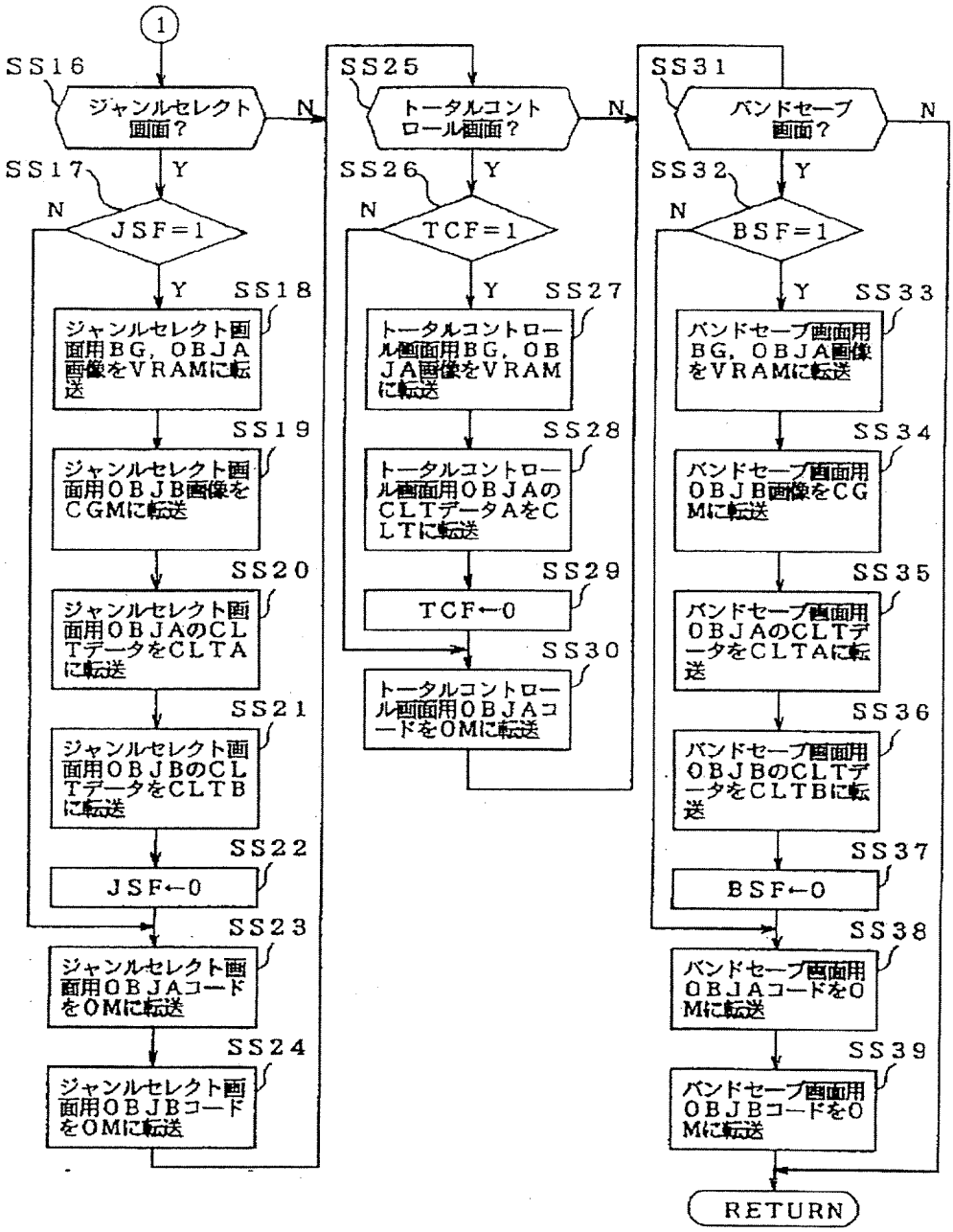
【図４２】


カラーブロック
変更処理（ＯＢＪＡ）
SR1
対象となるオブジェクト
を求める
SR2
ノーマル
N
SR3
対象となるオブジェクト
のノーマル色を持つカラ
ーブロック算出
SR4
対象となるオブジェクト
コードのカラーブロック
書き替え
SR5
N
セレクト？
SR6
対象となるオブジェクト
のセレクト色を持つカラ
ーブロック算出
SR7
対象となるオブジェクト
コードのカラーブロック
書き替え
RETURN

【図４３】


Vブランク
インタラプト
SS1
初期画面？
N
Y
SS2
IGF＝1
N
Y
SS3
初期画面用ＢＧ，ＯＢＪＡ画像をＶＲＡＭに転送
SS4
初期画面用ＢＧ用，ＯＢＪＡ用ＣＬＴデータをＣＬＴＡに転送
SS5
IGF←0
SS6
初期画面用ＯＢＪＡコードをＯＭに転送
SS7
バンド画面
N
Y
SS8
BGF＝1
N
Y
SS9
バンド画面用ＢＧ，ＯＢＪＡ画像をＶＲＡＭに転送
SS10
バンド画面用ＯＢＪＢ画像をＣＧＭに転送
SS11
バンド画面用ＢＧ用，ＯＢＪＡ用ＣＬＴデータをＣＬＴＡに転送
SS12
バンド画面用ＯＢＪＢ用ＣＬＴデータをＣＬＴＢに転送
SS13
BGF←0
SS14
バンド画面用ＯＢＪＡコードをＯＭに転送
SS15
バンド画面用ＯＢＪＢコードをＯＭに転送
1

【図44】



フロントページの続き

(51)Int.Cl.[6]		識別記号	庁内整理番号	FI		技術表示箇所
G09G	5/00	555	9377－5H	G09G	5/00	555D
	5/06		9377－5H		5/06	
	5/14		9377－5H		5/14	C
G10H	1/00			G10H	1/00	Z
H04N	7/18			H04N	7/18	P

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-081118

(43)Date of publication of application : 28.03.1997

(51)Int.Cl.

G09G 5/36
G09G 5/36
A63F 9/22
G06F 3/14
G09G 5/00
G09G 5/00
G09G 5/06
G09G 5/14
G10H 1/00
H04N 7/18

(21)Application number : 07-259254

(22)Date of filing : 11.09.1995

(71)Applicant : CASIO COMPUT CO LTD

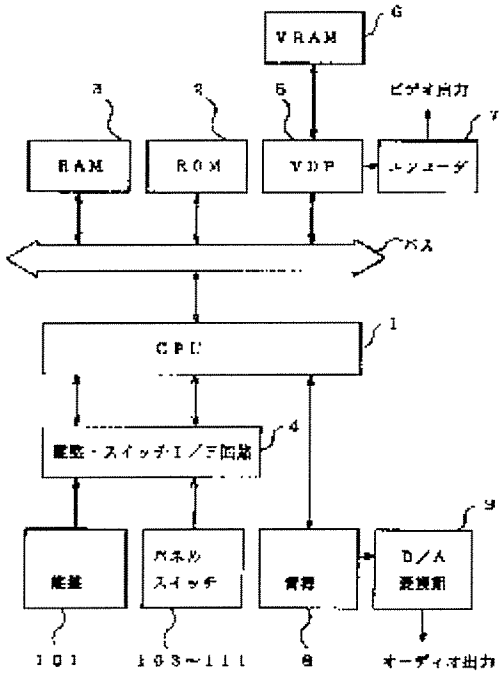
(72)Inventor : ISHIGURO SHIRO

(54) IMAGE CONTROL DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To improve the processing speed without using a high-speed CPU by making only the image a target of updating, of which a display mode is changed.

SOLUTION: In a character string alteration processing, 18 character strings, 6 lines of OBJB code, are read from a storage area of ROM 2 according to a cross key operation as the reference to a cursor position scrolled in accordance with value of a pointer register i, and when the read OBJB code is a space (blank), a transparent character (transparent color designation) is set, when it is not the case, the object B character (character image) specified by the OBJB code is set. Namely, with all the characters altered in display set to object B image, a corresponding OBJB code is designated in the case of altering display, and a transparent color may be designated in the case of erase display. Thus, a high speed image display is made possible.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] An image arrangement means to arrange the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled on the screen, and can be displayed in this viewing area, A modification means to specify either of the modification images arranged in said viewing area, and to change the display mode, The image control unit characterized by providing a display-control means to direct a screen display for the modification image which updated and updated the modification image according to the display mode changed by this modification means according to arrangement within said viewing area.
[Claim 2] The 1st attribute data which specifies the class, display position, and foreground color of the fixed image displayed fixed on a screen, An attribute data storage means to memorize the 2nd attribute data which specifies the class, display position, and foreground color of the modification image by which a display change is made on a screen, An image data storage means to memorize the image data matched with said 1st and 2nd attribute data, respectively, reading appearance of the image data which corresponds from said image data storage means according to said 1st and 2nd attribute data being carried out, respectively, and said modification image with an image display directions means to direct a screen display in piles, on said fixed image The rewriting means which rewrites said 2nd attribute data according to the directions when the directions which make a display change of said modification image are received, The image control unit with which only the modification image corresponding to said 2nd attribute data rewritten by this rewriting means is characterized by providing a display-control means to direct display modification.
[Claim 3] Said rewriting means is an image control unit according to claim 2 characterized by rewriting either the image class included in said 2nd attribute data, or a foreground color.
[Claim 4] Said rewriting means is an image control unit according to claim 2 characterized by rewriting to transparence the foreground color contained in said 2nd attribute data, and changing into a non-display condition when making a modification image into a null.
[Claim 5] Said 1st attribute data is an image control unit according to claim 2 characterized by specifying either of the usual fixed image with which it had the flag which specifies whether corresponding image data is supplied to said image display means in order of the address, or a reverse order is supplied, and the image display means concerned was supplied in order of the address according to the value of the flag concerned, and the reverse image which the reverse order was supplied and the image reversed.
[Claim 6] Said fixed image is an image control unit according to claim 2 characterized by having been formed from two or more image data, having changed the foreground color of said 1st attribute data matched for each [these] image data of every, and making a display mode adjustable.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention is used for example, for TV-game equipment etc., and relates to a suitable image control unit.
[0002]
[Description of the Prior Art] In recent years, it connects with AV (audio-visual) terminal of a television receiver, and various utilization of the TV-game equipment used as a game toy, an intellectual training toy, etc. is carried out. This kind of equipment is equipped with the image control unit which generally consists of CPU, a ROM, RAM, a VRAM (Video RAM), etc. While transmitting at VRAM each image data of the static image memorized by ROM and a dynamic image to the bottom of directions of CPU By reading each image data transmitted to this VRAM, changing this into a video signal, and supplying AV terminal of a television receiver The animation of two or more characters which move according to game actuation is displayed, displaying a background image (static image) on the Braun tube.
[0003]
[Problem(s) to be Solved by the Invention] By the way, in such an image control unit, although it is the translation which performs the same image control as a personal computer etc. fundamentally, it is necessary to make a product price possible the cheapest on the character in which it is used as a game toy or an intellectual training toy. For this reason, the hardware used inevitably will be restricted, for example, image memory (VRAM) capacity is stopped, and it is in the present condition of being hard to realize the operating environment called the so-called GUI (graphical user interface) environment, by some which omitted pointing devices, such as a mouse.
[0004] Moreover, in order to display a character string under the GUI environment, the character string displayed all over a window (display window) is treated as an object, and the character number and its display coordinate of each alphabetic character which should be displayed are set up separately. therefore, the character number and the display coordinate which correspond like a screen title also about the alphabetic character displayed fixed on a screen must be managed separately, and for actuation arrange the object of a character number which corresponds for erasing an alphabetic character on display further out of a viewing area is needed, the evil in_which disp of the case which displays a screen by these becomes slowly ****s [****] -- **. Although what is necessary is just to use high-speed CPU in order to raise processing speed, as mentioned above, a cost rise cannot be invited on product character, but this has been a technical technical problem.
[0005] Then, this invention was made in view of the situation mentioned above, and the main purpose is in offering the image control unit which realizes a high-speed screen display, without inviting a cost rise. Furthermore, as other purposes, image memory (VRAM) capacity is stopped and it is in offering the image control unit which can realize the false GUI environment also under the configuration which omitted pointing devices, such as a mouse.
[0006]
[Means for Solving the Problem] In order to attain the above-mentioned purpose, in invention according

to claim 1 An image arrangement means to arrange the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled on the screen, and can be displayed in this viewing area, A modification means to specify either of the modification images arranged in said viewing area, and to change the display mode, It is characterized by providing a display-control means to direct a screen display for the modification image which updated and updated the modification image according to the display mode changed by this modification means according to arrangement within said viewing area.

[0007] Moreover, the 1st attribute data which specifies the class, display position, and foreground color of the fixed image displayed fixed on a screen in invention according to claim 2, An attribute data storage means to memorize the 2nd attribute data which specifies the class, display position, and foreground color of the modification image by which a display change is made on a screen, An image data storage means to memorize the image data matched with said 1st and 2nd attribute data, respectively, reading appearance of the image data which corresponds from said image data storage means according to said 1st and 2nd attribute data being carried out, respectively, and said modification image with an image display directions means to direct a screen display in piles, on said fixed image The rewriting means which rewrites said 2nd attribute data according to the directions when the directions which make a display change of said modification image are received, Only the modification image corresponding to said 2nd attribute data rewritten by this rewriting means is characterized by providing a display-control means to direct display modification.

[0008] According to invention according to claim 3, as a desirable embodiment subordinate to above-mentioned claim 2, said rewriting means is characterized by rewriting either the image class included in said 2nd attribute data, or a foreground color.

[0009] Moreover, in invention according to claim 4, said rewriting means is characterized by rewriting to transparence the foreground color contained in said 2nd attribute data, and changing into a non-display condition, when making a modification image into a null.

[0010] It has the flag which specifies whether the image data to which said 1st attribute data corresponds at invention according to claim 5 is supplied to said image display means in order of the address, or a reverse order is supplied, and is characterized by specifying either of the usual fixed image with which the image display means concerned was supplied in order of the address according to the value of the flag concerned, and the reverse image which the reverse order was supplied and the image reversed.

[0011] Furthermore, in invention according to claim 6, said fixed image is characterized by having been formed from two or more image data, having changed the foreground color of said 1st attribute data matched for each [these] image data of every, and making a display mode adjustable.

[0012] If the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled in this invention on a screen, and an image arrangement means can display in this viewing area is arranged If either of the modification images arranged in a viewing area is specified and the display mode is changed with a modification means, a modification image will be updated according to the changed display mode, and a screen display of this will be carried out according to arrangement within said viewing area by the display-control means. That is, since only the image with which a display mode is changed serves as a candidate for updating, it becomes possible to aim at improvement in processing speed, without using high-speed CPU, and if it puts in another way, improvement in processing speed will be attained, without inviting a cost rise.

[0013]

[Embodiment of the Invention] It connects with a television receiver and the image control device by this invention displays a performance gestalt on the Braun tube, carries out the automatic performance of the performance information with the gestalt chosen from them, or may be applied to the musical-sound control device which generates the musical sound according to keyboard operation like the usual musical instrument. Below, such a musical-sound control unit is made into an example, and is explained with reference to a drawing.

[0014] A. See outside an example and explain the appearance of an example with reference to drawing 1

first. Drawing 1 shows the appearance of the musical-sound control unit 100 with which the image control unit by this invention was applied. In this drawing, 101 is the keyboard in which the key switch was arranged for every key, and generates the on-off signal according to the ****** actuation by the player. 102-111 are switches arranged in a panel side, respectively, and an electric power switch for 102 to turn on and off the power source of this equipment 100 and 103 are sound-volume switches which adjust the sound volume of generating musical sound. 104 is the Il Tempo switch which adjusts performance Il Tempo at the time of an automatic performance or automatic accompaniment. The tone switch with which 105 assigns a predetermined tone for every performance PERT, and 106 are rhythm switches which set the rhythm class of rhythm PERT at the time of an automatic performance or automatic accompaniment.

[0015] 107 is a stop switch operated in case an automatic performance or automatic accompaniment is stopped. 108 is a song switch operated in case the class of automatic performance data is chosen. 109 is a cross-joint key which consists of four-directions keys, and in case the cursor by which a screen display is carried out to the display terminal DP side mentioned later is moved to four directions, it is operated. 110 is an escape key, and in case it carries out setting cancellation and returns the time and a cursor location, it is operated. 111 is an enter key, and in case it carries out a definite input, it is operated. SP is a built-in loudspeaker arranged in the face of panel of the body of equipment. OUT1 and OUT2 are the audio output terminals and video outlet terminals which were prepared in the back-in-panels side side, respectively. The audio output and video outlet which are outputted from these output terminals OUT1 and OUT2 are connected to a television receiver or the well-known display terminal DP, and an image and voice (musical sound) are reproduced.

[0016] B. Describe the whole example configuration with reference to the configuration (1) of the whole configuration, next drawing 2 of an example. In addition, in this drawing, the same number is given to the element which is common into the part shown by the appearance of drawing 1 . In drawing 2 , 1 is CPU and controls the display and control section which consists of components 5, 6, and 7 which detect the event corresponding to panel switch actuation or ****** actuation, and mention it later according to the detected contents of an event, and the musical-sound control section which consists of components 8 and 9 based on a switch scan.

[0017] 2 is ROM the color look-up table which changes into color data (RGB data) the object code data showing attributes, such as a display coordinate location of the object image data for carrying out image display besides the various control programs performed in this CPU1 and object image data and a class, the background image data which forms a screen background, and these image data, or the song data for an automatic performance is remembered to be. The main data stored in this ROM2 are explained in full detail later on.

[0018] 3 is RAM, various register area is prepared as a work area of the above CPU 1, and the result of an operation, a flag value, etc. are stored temporarily. 4 is a keyboard and a switch interface circuitry. the on-off operation of the various actuation switches 103-111 with which this interface circuitry 4 is arranged in a panel side (refer to drawing 1) -- or the switch event at the time of on-off operation of the key switch formed for every key of a keyboard 101 being carried out by ****** actuation is generated, and CPU1 is supplied.

[0019] 5 is a video display processor (it is hereafter described as VDP) which consists of various logical operation components, and functions as a well-known CRT controller. VDP5 is what bears the function which carries out a display control to the bottom of directions of CPU1. While carrying out the DMA transfer of the image data stored in ROM2 to the character generator memory (it mentions later) or below-mentioned VRAM6 inside self Display-control processing which extracts the image data which should be displayed [from] among the image data stored in each [these] memory, and defines that display gestalt and display position is performed, and the image data to which this processing was performed is changed and outputted to the RGB data showing a foreground color. In addition, this configuration of VDP5 is described later on.

[0020] 7 is an encoder, superimposes a perpendicular/Horizontal Synchronizing signal on the RGB data outputted from VDP5, and generates a composite video signal. The image the display control was

carried out [the image] by VDP5 is displayed on CRT by supplying this composite video signal to a television receiver or the image input terminal of a display terminal DP. 8 is a sound source which consists of well-known wave memory read-out methods, reads the data point which corresponds from wave memory based on performance information, such as Keown / key-off when CPU1 occurs according to ****** actuation, or a velocity, or the automatic performance information (song data) read from ROM2 which CPU1 mentioned above, and generates musical-sound data. 9 is a D/A converter which changes and outputs the musical-sound data outputted from a sound source 8 to the audio output signal of analog format. Sound emission of the audio output signal outputted from this D/A converter 9 is carried out as a musical sound from the loudspeaker by the side of a display terminal DP through the output terminal OUT1 which sound emission was carried out from the built-in loudspeaker SP, or was mentioned above.

[0021] (2) Explain and go about the data configuration of ROM2, next the configuration of the main data stored in ROM2 with reference to drawing 3 - drawing 16 . Look-up table storage area ECLT the color look-up table which changes into color data (RGB data) the image data storage area EIMAGE where the background image data which forms the object image data and screen background corresponding to object storage area EOBJ the object code data which express with ROM2 the attribute, i.e., the object class, and its display coordinate location of the object which forms various screens are remembered to be, and these object codes is memorized, and each image data is remembered to be is prepared. Hereafter, the data configuration of these storage areas is described.

[0022] ** Explain the configuration of object storage area EOBJ with reference to the block diagram 3 of object storage area EOBJ - drawing 4 . In this storage area EOBJ, the object code for every various screens mentioned later assigns, and is memorized. Object code consists of two kinds of the objects B by which it is indicated by migration on a screen according to actuation of the object A displayed fixed on a screen or the above-mentioned cross-joint key 109, or an enter key 111. That is, only code OBJ1OBJA for object A and OBJ5OBJA which are displayed fixed on a screen, respectively are memorized by the object code OBJ1 for initial screens, and the object code OBJ5 for total control screens. In the object codes OBJ2, OBJ3, OBJ4, OBJ6, and OBJ7 corresponding to other screens, two kinds of codes of the object A displayed fixed on a screen and the object B by which it is indicated by migration are memorized.

[0023] Here, with reference to drawing 5 and drawing 6 , the data format of the object A code OBJA stored with the above-mentioned gestalt and the code OBJB for object B is explained. As for an OBJA/OBJB code, 1 word forms one object attribute by 3 words of WORD W0-W2 of 16 bit length. Magnitude (field), a display-position coordinate, a character name, a colour block (it mentions later) to be used of corresponding image data are expressed as an object attribute.

[0024] 9 bits (a bit 0 - bit 8) of low order of WORD W0 and W1 of the OBJA code / the OBJB code express the display-position coordinate on an object side (X, Y) so that it may illustrate to drawing 5 or drawing 6 . An object side is a dot flat surface defined by the coordinate field of - (0 0) (511,511). In the object side OBJA (or OBJB), 336 dots wide which shares the above-mentioned zero (0 0), and a 224 dots (scan line) long field serve as the screen. In addition, the display-position coordinate (X, Y) expresses the location of the upper left corner in an object field. That object becomes non-display when the object location defined by this display-position coordinate (X, Y) is not contained in the field of the above-mentioned screen.

[0025] The reversal flag X showing whether corresponding image data is reversed is set to the bit 9 of WORD W1 in the OBJA code. It specifies whether this flag X carries out the order of the write-in sequence at the time of storing object image data in a line buffer of the address, or it is made it and a reverse order, the order of the address is specified at the time of "0", and a reverse order is specified at the time of "1." Therefore, influencing [of an image] becomes reverse [the usual image written in in order of the address, and the image written in the reverse order], and both images become mirror image relation. that is, it is possible to increase the class of image which can be displayed [which shall usually boil one image data and shall inverse-video-display it] in a system with little memory space by being able to form two images with a thing and doing in this way.

[0026] object area size -- a minimum of 8 -- it is formed by dot x8 dot and adjustable assignment of the size (X size, Y size) with the field in every direction in 4 bits (a bit 12 - bit 15) of high orders of WORD W0 and W1 may be carried out. For example, when X size is "0" and the breadth of an object is 8 dots and "1", it becomes the greatest 128-dot width of face at 16-dot width of face and the time of "15." In WORD W2, the colour block of the color look-up table section 65 which 4 bits (a bit 12 - bit 15) of the high order mention later is specified, the class of foreground color is decided, and the character name showing the character class of object is set to 12 bits of low order after a bit 12. In addition, a colour block means what blocked the color look-up table CLT mentioned later for every predetermined color number, and assigned the color code.
[0027] ** In the configuration image data storage area EIMAGE of the image data storage area EIMAGE, like storage area EOBJ mentioned above, as shown in drawing 7 , the image data for every various screens assigns, and is memorized. Image data is classified into three kinds such as the background image data which forms a screen background, the object A image data as which a class and a display coordinate location are specified in above-mentioned object A code, and the object B image data as which a class and a display coordinate location are specified in object B code. That is, either of the three above-mentioned kinds of image data is assigned, and the image data IM1-IM5 of each screen forms a screen as illustrated to drawing 7 . Three kinds of image data forms BG screen, an OBJA screen, and an OBJB screen, respectively, and these screens overlap and it forms the one display screen. In the case of this example, the highest priority is given to an OBJB screen, then it becomes the order of an OBJA screen and BG screen. Therefore, as sequence of lapping, an OBJB screen is arranged most in this side, and it is arranged in order of an OBJA screen and BG screen after that to a back side.
[0028] Object B image data is formed in the cel unit which consists of 8 dots of lengthwise directions, and 8 dots of longitudinal directions so that it may illustrate to drawing 8 . A 1-bit color code is assigned, and when it is "0", it becomes a transparence code at each dot which constitutes one cel. 1 word is expressed 4 words on account of WORD C0-C3 of 16 bit length as the color code for one cel is shown in drawing 9 . That is, the color code of the dot lines d0-d7 in 1 cel is stored in 8 bits of high orders of WORD C0, and the following dot lines d10-d17 are stored in 8 bits of low order of WORD C0. Henceforth, each dot lines d20-d27, d30-d37, --, d60-d67, and d70-d77 are similarly stored in WORD C1-C3 serially.
[0029] On the other hand, although object A image data is formed like the above-mentioned object B image data considering 8 dots of 8 dot x longitudinal directions of lengthwise directions as 1 cel unit so that it may illustrate to drawing 10 , a 4-bit color code is assigned to each dot in this case, and it serves as an image containing a transparent plane color carried out 16 color specification. 1 word is expressed 16 words on account of WORD D0-D15 of 16 bit length as the color code for one cel which forms Object OBJA is shown in drawing 11 . That is, the color code of the dots d00-d03 in 1 cel is stored in WORD D0, and the following dots d04-d07 are stored in WORD D1. Henceforth, each dots d10-d13, d14-d17, --, d74-d77 are similarly stored in WORD D1-D15 serially.
[0030] ** In configuration look-up table storage area ECLT of look-up table storage area ECLT, the color look-up table corresponding to the image data for every various screens is memorized as illustrated to drawing 12 . That is, the CLT data CLT1 for initial screens are divided into image data IM1BG for initial screens mentioned above, look-up table CLT1BG for BG screens which changes IM1OBJA into color data, respectively, and look-up table CLT1OBJA for OBJA screens, and such its a table gestalt is the same also in other CLT data CT2-CLT5 for screens.
[0031] Here, with reference to drawing 13 - drawing 16 , the example of representation of a color look-up table is explained. First, drawing 13 is drawing showing the table format of look-up table CLT2OBJA for OBJA screens in the CLT data CLT2 for band screens. This table CLT2OBJA is equipped with 16 kinds of look-up tables specified by colour block number #0-#15, and the color data D1-D15 of 15 classification by color are stored in each table. Therefore, color display of the image data used as the read-out address of this table may be carried out by a maximum of 16 color containing a transparent plane color.
[0032] Such a table is assigned for each [which forms the OBJA screen in a band screen] object OBJA

of every. That is, it is specified with the "colour block value" (refer to drawing 5) stored in 4 bits of high orders in WORD W2 of the OBJA code mentioned above. In this table CLT2OBJA, the table which displays a band member's portrait image (it mentions later) on colour block number #0-#11 is assigned, and the title foreground color, the alphabetic character color, etc. are assigned to the colour block number #12-#15 remaining.

[0033] Next, drawing 14 is drawing showing the table format of look-up table CLT3OBJA for OBJA screens in the CLT data CLT3 for genre selection screens. As shown in this drawing, the object OBJA, i.e., a window, (display window) displayed fixed in the genre selection screen mentioned later, the window title, the scrolling foreground color, etc. are assigned as a colour block according to individual, respectively. On the other hand, table CLT3OBJB shown in drawing 15 is assigned to the object OBJB by which it is indicated by migration in a genre selection screen. In this case, since Object OBJB is the "alphabetic character" in which a 1-bit color code is assigned, "black" is assigned to colour block number #1 and "pink" is assigned to colour block number #2. In addition, it becomes transparence when colour block number #0 is assigned.

[0034] Drawing 16 is drawing showing the table format of look-up table CLT4OBJA for OBJA screens in the CLT data CLT4 for total control screens. As shown in this drawing, the colour block is assigned so that being "Normal", it being "a selection", and "it is active" may indicate the object OBJA which forms the "carbon button" displayed fixed in the total control screen mentioned later by the tri-state. In addition, "Normal", "a selection", and an "active" tri-state display are explained later on.

[0035] (3) Explain the configuration of VDP5 with reference to the configuration, next drawing 17 of VDP5. In this drawing, 50 is a CPU interface circuitry which is connected to a CPU bus, and delivers and receives CPU1 and data. 51 is a VRAM interface which manages data transfer with VRAM6. Background image data and object A image data are written in VRAM6 respectively through these interface circuitries 50 and 51 under directions of CPU1. Moreover, when reading these image data from VRAM6, like the time of writing, it reads from the CPU1 side through the above-mentioned interface circuitries 50 and 51, the address is given, and the image data read according to this is supplied to the below-mentioned controllers 59 and 63.

[0036] 52 is object memory in which the OBJA code for several object minutes displayed on a screen and the OBJB code are stored. The OBJA code and the OBJB code are data showing each attribute of the object A image data which forms Object OBJA, and the object B image data which forms Object OBJB, as illustrated to drawing 5 and drawing 6 . 53 is a control register which determines the mode of operation of VDP5. A control register 53 is a register of all 16 bit length, and a display-control gestalt is set up according to the register value set to each bit position by CPU1.

[0037] 54 is character generator memory and the object B image data which forms Object OBJB is memorized. The image data stored in this memory 54 forms an "alphabetic character", and as shown in drawing 8 , it is formed in the cel unit which consists of 8 dots of lengthwise directions, and 8 dots of longitudinal directions. A 1-bit color code is assigned, and when it is "0", it becomes a transparence code at each dot which constitutes one cel. Although the object A image data which forms the object OBJA stored in VRAM6 mentioned above on the other hand is formed like the above-mentioned object B image data considering 8 dots of 8 dot x longitudinal directions of lengthwise directions as 1 cel unit as shown in drawing 10 , a 4-bit color code is assigned to each dot in this case, and it serves as an image containing a transparent plane color in which 16 color specification is possible.

[0038] 55 is a DMA controller which carries out DMA transfer control according to directions of CPU1. DMA controller 55 carries out block transfer of either the data for a transfer, i.e., background image data, Object A, B image data and an OBJA/OBJB code to either VRAM6 and the object memory 52 used as a transfer assignment place, or the character generator memory 54 from ROM2 mentioned above according to the transfer directions from CPU1. 56 is an object lead controller, it is read from the object memory 52 which mentioned above the object attribute corresponding to the horizontal scanning Rhine location by the side of a display, sorts the read object attribute in the sequence according to the quota gestalt of the object class specified by a control register 53, and stores it in the stack B memory 58 and the stack A memory 57 which mention the result later.

[0039] That is, by this controller 56, the OBJA/OBJB code of the Y coordinate which agrees in that scan line location whenever horizontal scanning Rhine is updated, and Y size is extracted from from among the OBJA code / the OBJB code stored in the object memory 52. And it asks for the display order on 1 horizontal-scanning Rhine based on the X coordinate of an OBJA/OBJB code and X size which were extracted, and the display priority specified by a control register 53. Next, in the sequence searched for, the OBJA code is stored in the stack A memory 57, and the OBJB code is stored in the stack B memory 58, respectively.
[0040] Therefore, by the stack A memory 57 and the stack B memory 58, the OBJA code which the object attribute which may be displayed in 1 horizontal-scanning period, respectively, i.e., an OBJA/OBJB code, will be memorized, and is memorized by these memory 57 and 58, and the OBJB code are read in order of a display by the controllers 59 and 60 mentioned later. The line buffer A controller 59 reads in order the OBJA code memorized in order of the display in the above-mentioned stack A memory 57, and reads the object A image data corresponding to the "character name" contained in the OBJA code concerned from VRAM6.
[0041] Moreover, this controller 59 writes the object A image data which read the X coordinate value included in the OBJA code from VRAM6 as the write-in address in a line buffer A61 (it mentions later). In case object A image data is written in a line buffer A, when the reversal flag X located in the bit 9 of WORD W1 in the OBJA code is "1", the write-in address is made into a reverse order, and when the reversal flag X is "0", on the other hand, it writes in in order of the address. Thereby, it may have comes to generate two mirror image-related images which right and left of an image reverse. Moreover, the colour block value (refer to drawing 3) extracted from the OBJA code is given to the object A image data written in a line buffer A61.
[0042] The line buffer B controller 60 reads in order the OBJB code memorized in order of the display in the stack B memory 58, reads the object B image data corresponding to the "character name" contained in the OBJB code concerned from the character generator memory 54, and writes it in a line buffer B62 (it mentions later). In the case of this writing, the X coordinate value in the OBJB code is used as the write-in address. Moreover, the "colour block" (refer to drawing 4) extracted from the OBJB code is given to the object B image data written in a line buffer B62.
[0043] The line buffer memory which stores temporarily the image data for 1 horizontal-scanning Rhine, respectively is prepared in the line buffer A61 and the line buffer B62 by two or more lines, and it is constituted so that the image data for the scan line displayed at least synchronizing with the horizontal scanning of a degree besides the image data for the scan line displayed synchronizing with a current horizontal scanning may also be stored temporarily. Moreover, it is controlled by the line buffer A61 and the line buffer B62 by the controllers 59 and 60 mentioned above so that buffer memory might be switched by turns and it might Rhine-read or Rhine write in synchronizing with a horizontal scanning. That is, apparent processing speed is improved by writing character data in the buffer memory of another side, reading character data from one buffer memory.
[0044] 63 is the BG code lead controller, reads the background image data stored in VRAM6, and stores it in the latter BG data register 64. The background image data stored in this register 64 serves as a static image which forms a screen background. While 65 gives screen priority, it is the color look-up table section which changes and outputs the image data of each screen where priority was given to color data (RGB data). Priority is given to each image data (Object A, B image data, and background image data) first outputted synchronizing with a horizontal scanning from the line buffer A61, the line buffer B62, and the BG data register 64 which were mentioned above in this table section 65. The priority showing the stacking-order foreword of a screen becomes settled according to the register value of a control register 53 mentioned above.
[0045] In the table section 65, it is divided into storage area CLTA which memorizes the color look-up table data corresponding to an object A image, and storage area CLTB which memorizes the color look-up table data corresponding to an object B image, and the DMA transfer of the color look-up table data of the class specified with the "colour block value" included in the object code corresponding to each image data is carried out to each [these] storage area. And this table section 65 chooses the color look-

up table of a colour block number which corresponds according to the "colour block value" in the object code corresponding to each image data to which screen priority was given, and changes and outputs each image data to color data (RGB data) based on the selected table.
[0046] 66 is a D/A converter, and changes and outputs the color data (RGB data) outputted from the above-mentioned color look-up table section 65 to the chrominance signal for every color RGB. In addition, each chrominance signal is superimposed on a horizontal/Vertical Synchronizing signal in the encoder 7 mentioned above, and turns into a composite video signal. 67 is the synchronizing signal generating section which carries out the multiplying oscillation of the original ** clock of crystal oscillator X'tal, generates a horizontal synchronization / vertical-synchronization clock, and supplies this to each part in VDP5.
[0047] (4) Explain a screen configuration, next the configuration of the various screens which VDP5 forms based on the various data stored in ROM2. As mentioned above, to ROM2 An "initial screen", a "band screen", a "band member modification screen", Although it is the translation in which the object code, the image data, and look-up table data for forming a "genre selection screen", a "total control screen", and a "band save screen" are stored Here, the configuration of a "band screen", a "band member modification screen", a "genre selection screen", and a "total control screen" is described as a typical thing among these.
[0048] ** The configuration band screen of a band screen is located in this viewing-area [which is displayed on a display] A, and viewing-area A bottom, and consists of viewing areas B which serve as invisibility, without being displayed on a display. In addition, drawing 18 illustrates one mode of a viewing area A. In this viewing area A, parts which do not carry out display change, such as a title bar, and a title alphabetic character, a selection carbon button or a musician's portrait image, are formed by the object A image (OBJA), and, on the other hand, the character string CHR1 which carries out display change on this object A image according to actuation of the cross-joint key 109 or an enter key 111 - the CHR8 grade are formed by the object B image (OBJB).
[0049] In a viewing area B, when it is evacuated as an object A image and display selection of the selection carbon button, title bar, and portrait image by which display use is not carried out in the viewing area A is made according to selection actuation of the cross-joint key 109, the display coordinate location in the OBJA code is rewritten, and it moves to a viewing-area A side, and displays. moreover, color look-up table CLT2OBJA (refer to drawing 13) which mentioned above each object A image put on a viewing-area A side -- each -- it is matched with colour block #0-#15, and is expressed as the color data in an assignment colour block.
[0050] ** The configuration, next drawing 19 of a band member modification screen are drawing showing the band member modification screen displayed when it is operated so that a member change may be made in an above-mentioned band screen. While moving the selections display area displayed on the lower half of a band screen to a viewing area B, the portrait image for 3 persons is instead chosen and expressed as a band member modification screen from from among the portrait images for two or more person minutes of several arranged at the viewing area B. Drawing 19 illustrates such a condition. By the way, the portrait images P1 and P2 shown in drawing 19 are formed of the same object A image data, and two mirror image-related images with which right and left of an image reverse the reversal flag X in the OBJA code mentioned above when generating the image of one of these by setting to "1" have been obtained. In addition, also in such a band member modification screen, the part which does not carry out display change is formed by the object A image (OBJA), and the character string which carries out display change is formed by the object B image (OBJB).
[0051] ** The window WIN used in case a performance genre is chosen as middle of the screen, as shown in drawing 20 is expressed as the configuration genre selection screen of a genre selection screen. This window WIN consists of object OBJA-50 which display object OBJA-1 - OBJA8, and the condition that can be scrolled which constitutes a title alphabetic character, a title bar (object OBJA-52) by which OBJA-51 are arranged, and a color adjustable carbon button (object OBJA-51-OBJA-60) of eight lines so that it may illustrate to drawing 21 .
[0052] With each carbon button (object OBJA-51-OBJA-60), 18 trains of objects OBJB of the 8 dot x 8-

dot size of ends of a road are arranged. That is, matrix arrangement of object OBJB-128-OBJB-271 is carried out. These object OBJB-128-OBJB-271 form an alphabetic character, or the rarefaction is carried out, and they are set as display ranking higher than the above-mentioned object OBJA-51-OBJA-60. Therefore, it seems that the character string is drawn on the carbon button. Moreover, each carbon button may change a foreground color, as it is shown by changing the colour block number of color look-up table CLT3OBJA (refer to drawing 14) mentioned above according to the vertical key stroke of the cross-joint key 109 whether it is in a selection condition.

[0053] ** Also in the band save screen shown in the block diagram 22 of a band save screen, the same window WIN as **** is arranged. In the window WIN in a van save screen, the class of object OBJB which forms the alphabetic character on each carbon button is changed, and a character string can be rewritten now. In that case, the foreground color of Object OBJB changes the colour block number of color look-up table CLT3OBJB shown in drawing 15 to #2 (pink) from #1 (black), rewrites it, and can indicate that it is inside.

[0054] ** The configuration, next drawing 23 of a total control screen are drawing showing an example of a total control screen. On this screen, the "carbon button" which sets up the terms and conditions of equipment 100 of operation is formed of Object A. Each carbon button is constituted so that it may be in the tri-state of the active state (c) which is visible to convex [which seem to illustrate to the drawing 24 (**) in a concave / the Normal condition (a), the selection condition (b), and convex]. That is, by OBJA-D and E which form the alphabetic character on OBJA-A which forms the left brink upper limb of a carbon button, OBJ-B which forms the right-margin-of-heart margo inferior of a carbon button, OBJA-C which forms a carbon button center section, and a carbon button, a carbon button is formed, the foreground color of these object OBJA-A-OBJA-C is changed according to the contents of setting actuation, and the condition of above-mentioned (a) - (b) is expressed as shown in the drawing 24 (**).

[0055] For example, "RESUME" in drawing is explained as an example. First, with the carbon button of a non-established state, the color data D13, D14, and D15 in colour block number #0 of color look-up table CLT4OBJA are assigned to above-mentioned object OBJA-A-OBJA-C, respectively, and it changes into the Normal condition (a) which is visible to a concave. When carbon button selection is made by the right-and-left key stroke of the cross-joint key 109 from this condition, the color data D13, D14, and D15 of colour block number #8 are assigned to object OBJA-A-OBJA-C, a foreground color is changed, and a selection condition (b) is expressed. And if an enter key 111 is pushed and a setup is made to decide, the color data D13, D14, and D15 of colour block number #7 will be assigned to object OBJA-A-OBJA-C, and it will be made the active state (c) which is visible to convex. It becomes possible to realize a GUI operating environment to which is not equipped with pointing devices, such as a mouse, but ** also clicked the carbon button by doing in this way.

[0056] C. Explain actuation of an example, next actuation of the example by the above-mentioned configuration with reference to drawing 25 - drawing 43 . After describing the outline of whole actuation first, about the detail of the image processing in connection with the summary of this invention, sequential explanation is given and it goes by below.

[0057] (1) An injection of the electric power switch 102 of the equipment 100 which is whole (actuation a) outline flow this example operates CPU1 according to the flow shown in drawing 25 . That is, if an electric power switch 102 is switched on, it progresses to a step SA 1, and it will progress to a step SA 2, after carrying out zero reset of the register and flag which are formed in each part of RAM3, VRAM6, VDP5, and a sound source 8 or initializing setting initial value etc. At a step SA 2, while scanning a keyboard 101 and a panel switch group, detecting a ****** actuation event or a switch actuation event, generating musical sound or performing pronunciation processing / music processing which sets the various performance parameters at the time of carrying out musical-sound generating according to the detected event, the image processing for generating a screen display corresponding to these processings is performed.

[0058] Subsequently, if a step SA 2 is completed, CPU1 will advance processing to a step SA 3, and it will judge whether an auto-power-off setup was carried out. Here, if it is not in a power-off condition, a decision result serves as "NO", processing is returned to a step SA 2, the Maine flow processing is

continued, on the other hand, when a power-off setup is carried out, a decision result will serve as "YES" and processing will be advanced to the following step SA 4. At a step SA 4, after, performing resume processing which carries out memory evacuation of each part established state in equipment 100, silence processing which carries out mute of the musical sound under pronunciation for example, power-off processing which carries out power-source OFF is performed, and a halt of operation is carried out.
[0059] (b) If processing of the Maine flow CPU 1 progresses to the above-mentioned step SA 2, the Maine flow shown in drawing 26 will be performed, and it will progress to a step SB 1. At a step SB 1, a keyboard scan is performed that ****** actuation of a keyboard 101 should be detected. Then, at a step SB 2, while generating performance information, such as Keown, key-off, a keycode, and a velocity, based on the ****** actuation event detected in this keyboard scan, pronunciation processing which directs musical-sound generating in a sound source 8 according to this performance information is performed. Subsequently, at a step SB 3, in order to detect switch actuation of panel switches 103-111, a panel switch scan is performed and a switch actuation event is generated.
[0060] Next, at a step SB 4, the switch actuation which creates the automatic performance information on time series among the generated switch actuation events is extracted, and an event matrix is created. It becomes the performance pattern which will serve as this event matrix from a pitch and pronunciation timing when becoming the rhythm pattern which enumerated pronunciation timing and creating automatic performance information if automatic rhythm performance information is created. Subsequently, if it progresses to a step SB 5, music processing which sets up the effectiveness gestalt given when CPU1 generates the rhythm pattern or performance pattern expressed as the above-mentioned event matrix as a musical sound will be performed. Then, that the screen corresponding to the image processing in connection with a summary, i.e., the above-mentioned switch actuation event, of this invention should be generated, a display control is carried out, and at the continuing step SB 7, CPU1 performs that alien-system processing, and completes this Maine flow.
[0061] (2) Explain actuation of an image processing, next the contents of the image processing performed in the above-mentioned step SB 6 in full detail. If processing of CPU1 progresses to the above-mentioned step SB 6, the image-processing routine shown in drawing 27 will be started, and processing will be advanced to a step SC 1. At a step SC 1, it judges whether the switch actuation event which changes whether there is any mode change and a screen display that is, occurred. Here, if there is a switch actuation event which changes a screen display, a decision result serves as "YES", processing is advanced to the following step SC 2, and when that is not right, processing will be advanced to the below-mentioned step SC 3. If it progresses to a step SC 2, CPU1 will perform screen change processing which extracts the object code which corresponds from ROM2 according to the contents of a screen which should be changed, and is stored in the object memory 52 of the VDP5 interior. In addition, about the detail of screen change processing, it mentions later.
[0062] Subsequently, at steps SC3-SC6, the display control corresponding to the switch actuation event generated under each screen mode is performed. That is, if it is in a step SC 3, the display mode of the character strings CHR1-CHR8 displayed all over a band screen (refer to drawing 18) according to actuation of the cross-joint key 109 mentioned above, or an escape key 110 or an enter key 111 is changed. Next, at a step SC 4, the character string in the window WIN displayed on a genre selection screen (refer to drawing 20) according to actuation of the cross-joint key 109 or an enter key 111 is indicated by migration, or the display control which makes a color change is performed. And at a step SC 5, the display control which changes the foreground color of the "carbon button" for every item in a total control screen (refer to drawing 23) according to switch actuation is performed. Furthermore, on a band save screen (refer to drawing 22), according to actuation of the cross-joint key 109 or an enter key 111, the character string in Window WIN is indicated by migration, or the display control which makes a color change is performed in a step SC 6.
[0063] About the detail of the display control made in these steps SC3-SC6, it mentions later. Moreover, CPU1 which performs the display control by these steps SC3-SC6 carries out interrupt processing of the V blank interrupt manipulation routine (it mentions later) for every fixed period, and updates the screen by which the display control was carried out. That is, in the process in which steps SC3-SC6 are

performed, CPU1 performs V blank interrupt manipulation routine synchronizing with the vertical-retrace-line period by the side of Display DP, and carries out the DMA transfer of object code required for a screen display, image data, and the color look-up table to the VDP5 and VRAM6 side. And in VDP5, a screen is updated based on the various above-mentioned data by which the DMA transfer was carried out in this way to the character generator memory 54 inside VRAM6 or self, and the object memory 52.

[0064] (a) Explain actuation of a screen change manipulation routine, next actuation of a screen change manipulation routine for every screen.

** If the switch actuation event to an initial screen which changes a screen display as changed and mentioned above occurs, CPU1 will perform the screen change manipulation routine shown in drawing 28 through a step SC 2, and will advance processing to a step SD 1. At a step SD 1, the generated switch actuation event judges whether it is what directs the change to an initial screen. Here, if it is the event which directs the change to an initial screen, the decision result of a step SD 1 will serve as "YES", and, as for CPU1, processing will be advanced to a step SD 2. At a step SD 2, the value of the initial-screen transfer flag stored in Register IGF is set to "1."

[0065] Subsequently, if it progresses to a step SD 3, the object code OBJ1 (refer to drawing 3) for initial screens stored in ROM2 will be stored in the work area of RAM3. By doing in this way, during a vertical blanking interval, the background image data, the object A image data, and the CLT data CLT1 for initial screens which form an initial screen are transmitted to the VRAM6 and VDP5 side by V blank interrupt manipulation routine mentioned later, and, thereby, an updating indication of the initial screen is given.

[0066] ** When the switch actuation event which directs a change on the change band screen to a band screen occurs, the decision result of the above-mentioned step SD 1 serves as "NO", advance processing to a step SD 4, and judge whether it is a change on a band screen. And the decision result of a step SD 4 serves as "YES", and advances processing to the following step SD 5 because a change on a band screen is performed in this case. At a step SD 5, the value of the band screen transfer flag stored in Register BGF is set to "1." Subsequently, if it progresses to a step SD 6, code OBJ2OBJA for object A will be extracted from from among the object codes OBJ2 (refer to drawing 3) for band screens stored in ROM2, and it will store in the work area of RAM3.

[0067] Then, if it progresses to a step SD 7, CPU1 will extract code OBJ2OBJB for object B from from like the above-mentioned step SD 6 among the object codes OBJ2 (refer to drawing 3) for band screens stored in ROM2, and will store it in the work area of RAM3. And in the below-mentioned V blank interrupt manipulation routine, image data IM2 (refer to drawing 7) for band screens and the CLT data CLT2 (refer to drawing 12) for band screens which form a band screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side by this, and a band screen is formed.

[0068] ** When the switch actuation event which directs a change on a genre selection screen, next a change on a genre selection screen occurs, all serve as "NO", and the decision result of the above-mentioned steps SD1 and SD4 advances processing to a step SD 8, and judges whether it is a change on a genre selection screen. And the decision result of a step SD 8 serves as "YES", and processing is advanced to the following step SD 9 because a change on a genre selection screen is performed in this case. At a step SD 9, the value of the genre selection screen transfer flag stored in Register JSF is set to "1." Subsequently, if it progresses to a step SD 10, code OBJ4OBJA for object A will be extracted from from among the object codes OBJ4 (refer to drawing 3) for genre selection screens stored in ROM2, and it will store in the work area of RAM3.

[0069] Then, if it progresses to a step SD 11, CPU1 will extract code OBJ4OBJB for object B from from like the above-mentioned step SD 10 among the object codes OBJ4 (refer to drawing 3) for genre selection screens stored in ROM2, and will store it in the work area of RAM3. Consequently, based on the below-mentioned V blank interrupt manipulation routine, image data IM3 (refer to drawing 7) and the CLT data CLT3 (refer to drawing 12) for genre selection screens which form a genre selection screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side, and a genre selection screen is updated.

[0070] ** If the switch actuation event which directs a change on the change total control screen to a total control screen occurs, all will serve as "NO", and the decision result of the above-mentioned steps SD1, SD4, and SD8 will advance processing to a step SD 12, and will judge whether it is a change on a total control screen. And since a change on a total control screen is performed in this case, the decision result of a step SD 12 serves as "YES", and advances processing to the following step SD 13.
[0071] At a step SD 13, the value of the total control screen transfer flag stored in Register TCF is set to "1." Subsequently, if it progresses to a step SD 14, the object code OBJ5 (refer to drawing 3) for total control screens stored in ROM2 will be stored in the work area of RAM3. Consequently, a total control screen is updated by transmitting image data IM4 (referring to drawing 7) and the CLT data CLT4 (referring to drawing 12) for total control screens which form a total control screen during a vertical blanking interval to the VRAM6 and VDP5 side based on the below-mentioned V blank interrupt manipulation routine.
[0072] ** If the switch actuation event which directs a change on the change band save screen to a band save screen occurs, all will serve as "NO", and the decision result of steps SD1, SD4, SD8, and SD12 mentioned above will advance processing to a step SD 15, and will judge whether it is a change on a band save screen. And the decision result of a step SD 15 serves as "YES", and processing is advanced to the following step SD 16 because a change on a band save screen is performed in this case. At a step SD 16, the value of the band save screen transfer flag stored in Register BSF is set to "1." Subsequently, if it progresses to a step SD 17, code OBJ6OBJA for object A will be extracted from from among the object codes OBJ6 (refer to drawing 3) for band save screens stored in ROM2, and it will store in the work area of RAM3.
[0073] Then, at a step SD 18, code OBJ6OBJB for object B is extracted from from like the above-mentioned step SD 17 among the object codes OBJ6 (refer to drawing 3) for band save screens stored in ROM2, and it stores in the work area of RAM3. Consequently, based on the below-mentioned V blank interrupt manipulation routine, image data IM5 (refer to drawing 7) and the CLT data CLT5 (refer to drawing 12) for band save screens which form a band save screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side, and a band save screen is updated. In addition, when other different events from the event which directs a screen change occur, as for each decision result in the above-mentioned steps SD1, SD4, SD8, SD12, and SD15, all serve as "NO", and CPU1 completes this routine.
[0074] (b) When events other than a screen change occur when a band screen manipulation routine carries out **** of operation and a screen change manipulation routine is completed or, CPU1 performs the band screen manipulation routine shown in drawing 29 through the step SC 3 mentioned above, and advances processing to a step SE 1. At a step SE 1, it judges whether a screen current on display is a band screen. Here, in not being the mode which displays a band screen, a decision result serves as "NO" and completes this routine. On the other hand, while displaying the band screen, a decision result serves as "YES" and processing is advanced to the following step SE 2. If it progresses to a step SE 2, as for CPU1, the generated event will judge whether it is the event which requires image modification.
[0075] The event which requires image modification points out band member modification actuation of replacing the portrait image of the case where the character string CHR1 in a band screen (refer to drawing 18) is chosen according to the vertical key stroke of the cross-joint key 109, and the musician displayed for every PERT. And when actuation which needs image modification is performed, the decision result of the above-mentioned step SE 2 serves as "YES", and processing is advanced to a step SE 3. On the other hand, when the actuation which does not need image modification is made, a decision result serves as "NO" and completes this routine.
[0076] Now, when actuation which needs image modification is performed, CPU1 advances processing to a step SE 3, and it judges whether it is that by which the generated event changes a character string. Here, if the upper key of the cross-joint key 109 or a bottom key is operated that either of the character strings CHR1 in a band screen (refer to drawing 18) should be chosen and a cursor location is scrolled, a decision result will serve as "YES" and processing will be advanced to the following step SE 4. At a step SE 4, the character string modification manipulation routine (it mentions later) which chooses either

of the character strings CHR1 according to this scrolling actuation is performed. Subsequently, at a step SE 5, the colour block number matched with the selected character string (object OBJB) is rewritten, and this routine is completed, after performing the colour block modification manipulation routine (it mentions later) which changes the foreground color of that character string.
[0077] On the other hand, when actuation other than character string modification is made when a decision result is "NO" in the above-mentioned step SE 3 namely, processing is advanced to a step SE 6. At a step SE 6, it judges whether the actuation in which it succeeded is actuation, i.e., band member modification actuation of replacing with other things either of a musician's portrait images displayed all over a band screen, of changing a band member. And if the actuation is performed, a decision result here will serve as "YES" and processing will be advanced to the following step SE 7. At a step SE 7, while displaying PERT's portrait image by which modification assignment was carried out on an assignment coordinate location, the band member modification manipulation routine (it mentions later) which changes the foreground color of the "PERT name" part arranged at the lower part of the portrait image made applicable to modification is performed, and this routine is completed.
[0078] (c) If the upper key of the cross-joint key 109 or a bottom key is operated that either of the character strings CHR1 in a band screen should be chosen and a cursor location is scrolled as the character string modification manipulation routine carried out **** of operation, CPU1 will perform the character string modification manipulation routine shown in drawing 30 through a step SE 4, and will advance processing to a step SF 1. The direction of operated scrolling is distinguished at a step SF 1. The scrolling direction progresses to a step SF 2, when the upper key of the cross-joint key 109 is operated a "top" that is, and the decrement of the direction address of Y which shows the display area on ROM2 is carried out by one line.
[0079] That is, code OBJ2OBJB for object B which forms the alphabetic character in which a migration display is possible all over a band screen is stored in the form corresponding to the display gestalt on a screen in object storage area EOBJ of ROM2. The storage area serves as direction of X (direction of train) 18 character string, and the direction (line writing direction) of six lines of Y, and a screen display of this is carried out as a character string CHR1. Therefore, when the upper key of the cross-joint key 109 is operated and it scrolls to the bottom on a screen, 1 decrement of the direction read-out address of Y of the OBJB code in the above-mentioned storage area is carried out. Moreover, when the bottom key of the cross-joint key 109 is operated and it scrolls to the bottom on a screen like this, based on distinction of the above-mentioned step SF 1, processing is advanced to a step SF 3 and the direction read-out address of Y is incremented one time in this case.
[0080] Subsequently, if it progresses to a step SF 4, CPU1 will read the present address (X, Y) updated according to scrolling actuation, and will use it as an initiation pointer. And "1" is set to the pointer register i at the following step SF 5. In addition, a sequential increment is carried out, and the value of this pointer register i is read, and is treated as a relative address on the basis of an initiation pointer so that it may mention later. and criteria [cursor location / which was scrolled henceforth / a step SF 6] -- carrying out -- 18 character string [from the above-mentioned storage area] x -- the OBJB code for six lines is read according to the value of the pointer register i. That is, at a step SF 6, it judges whether 18 character strings and the OBJB code for six lines were read. Here, when 18 character strings and the OBJB code for six lines are read, a decision result serves as "YES" and progresses to a step SF 7. At a step SF 7, a read-out initiation pointer is reset and this routine is completed.
[0081] On the other hand, when read-out of the OBJB code is not completed, the decision result of the above-mentioned step SF 6 serves as "NO", it progresses to a step SF 8, and the OBJB code train which forms the character string specified based on the value and the present address (X, Y) of the pointer register i is read. Subsequently, at a step SF 9, the read OBJB code train is interpreted a single character every, and it judges whether the interpreted OBJB code is "a tooth space (null)." Here, a decision result becomes being "a tooth space (null)" with "YES", processing is advanced to the following step SF 10, and it is set as a transparence character (transparence color specification).
[0082] On the other hand, when it is not "a tooth space (null)", the decision result of the above-mentioned step SF 9 serves as "NO", processing is advanced to a step SF 11, and it is set as the object B

character (alphabetic character image) specified by the OBJB code. And after this, CPU1 advances processing to a step SF 12, stores the interpreted OBJB code in the object memory 52 of the VDP5 interior, increments the pointer register i one time in the continuing step SF 13, and returns processing to the above-mentioned step SF 6.

[0083] It is based on the cursor location scrolled according to actuation of the cross-joint key 109 in character string modification processing. From the storage area of ROM2 to thus, 18 character strings The OBJB code for six lines is read according to the value of the pointer register i. When the read OBJB code is "a tooth space (null)", a transparence character (transparence color specification) is set up, and when that is not right, he is trying to set it as the object B character (alphabetic character image) specified by the OBJB code. That is, all the alphabetic characters by which a display change is made are set as the object B image, and since what is necessary is to specify the corresponding OBJB code and just to specify it as a transparent plane color in the case of an elimination display, in case a display change is made, compared with the conventional thing which specifies a display gestalt and a display coordinate location, it can grow into every character (alphabetic character) with a high-speed screen display.

[0084] (d) Explain actuation of a colour block modification processing (OBJB) routine, next actuation of the colour block modification manipulation routine which changes the foreground color of the object OBJB set as the object of display modification with reference to drawing 31 . An object B image can change a colour block number, and can change a foreground color to "black" and "pink" by turns since the 1 bit [per dot] color code is assigned as mentioned above, for example, so that color look-up table CLT3OBJB illustrated to drawing 15 may show. When changing the colour block of Object OBJB, "black" is equivalent to the below-mentioned "Normal color", and "pink" is equivalent to the below-mentioned "active color."

[0085] Suppose that modification of a character string (object OBJB) was completed by the now, for example, above-mentioned character string, modification manipulation routine. If it does so, CPU1 will perform the colour block modification processing (OBJB) routine shown in drawing 31 through the step SE 5 mentioned above in order to change the foreground color of this changed character string, and will advance processing to step SG1. At step SG1, the object first made applicable to modification is specified. Subsequently, at step SG2, it judges whether the foreground color currently assigned to the object made into the object is the "Normal color." And if it is the "Normal color", a decision result will serve as "YES", processing will be advanced to the following step SG3, and the colour block number which assigns an "active color" to the object will be computed.

[0086] On the other hand, when the foreground color currently assigned to the target object is an "active color", the decision result of the above-mentioned step SG2 serves as "NO", and CPU1 advances processing to step SG4, and computes the colour block number which assigns the "Normal color" to the object. And if it progresses to step SG5 after this, the colour block number in the target object OBJB code will be rewritten for the colour block number computed at the above-mentioned step SG3 or step SG4. Thus, in colour block modification processing (OBJB), when making a display change in the state of Normal, the colour block number of an active color is assigned, and in [that] being opposite, it assigns the colour block number of the Normal color.

[0087] (e) In the band screen of a band member modification manipulation routine of operation, if band member modification actuation of replacing with other things either of five kinds of a musician's portrait images displayed all over a screen is made, CPU1 will perform the band member modification manipulation routine shown in drawing 32 through the above-mentioned step SE 7 (refer to drawing 29), and will progress to step SH1. An object migration manipulation routine (it mentions later) is performed, and the specified location is made to indicate a member's portrait image by migration at step SH1. And at step SH2, the foreground color of the object OBJA corresponding to this portrait image by which it was indicated by migration is changed based on actuation of an object color modification manipulation routine (it mentions later).

[0088] (f) If the above-mentioned step SH1 of an object migration manipulation routine of operation is performed, CPU1 will perform the object migration manipulation routine shown in drawing 33 , and will

advance processing to step SJ1. At step SJ1, it judges whether the actuation which displays a selection member was made. The actuation which displays a selection member here moves cursor to the location of the character string CHR7 ("MEMBER") displayed on the lower part of the band screen illustrated to drawing 18 , and after pushing an enter key 111 there and making band member maintenance mode change, it points out actuation of specifying the "PERT" who should change. When such actuation is made, the decision result of the above-mentioned step SJ1 serves as "YES", and processing is advanced to the following step SJ2. on the other hand, the actuation which displays a selection member should do -- when there is nothing, it progresses to the below-mentioned step SJ4.

[0089] At step SJ2, the display-position coordinate (X, Y) of each object OBJA code which forms selections display area is changed so that it may move to the viewing area B which becomes the outside of a screen display rectangle. In addition, the selections display area said here points out the screen range of a lower half from the portrait image in a band screen. And in the following step SJ3, the display-position coordinate (X, Y) of the object OBJA code to the selected member is moved to a viewing-area A side as a substitute of the above-mentioned selections display area. In addition, the selected member means a selection candidate's portrait image beforehand assigned to the "PERT" set as the object of member modification. That is, in the above-mentioned step SJ2 - SJ3, if the actuation which displays a selection member is made, while moving selections display area to the viewing area B which becomes the outside of a screen display rectangle, the portrait image of the selection candidate stationed at the viewing-area B side is moved to a viewing-area A side as a substitute of selections display area. Thereby, the band member modification screen illustrated to drawing 19 is formed.

[0090] Next, if it progresses to step SJ4, CPU1 will judge whether actuation, i.e., the actuation on which the portrait image which serves as a new candidate is displayed, of changing a selection member was performed. And if actuation of changing a selection member is made, a decision result will serve as "YES", will advance processing to the following step SJ5, will move the display-position coordinate (X, Y) of a member's object OBJA code chosen by modification to a viewing-area A side, and will display as a new candidate's portrait image. Subsequently, at step SJ6, in response to a new candidate's portrait image being displayed, the display-position coordinate (X, Y) of a member's object OBJA code which separated by modification is moved to a viewing-area B side, and suppose that it is non-display.

[0091] The band member modification screen which illustrates actuation of the above-mentioned steps SJ5 and SJ6 to drawing 19 is mentioned as an example, and is explained. Suppose that three persons' portrait image which serves as a selection candidate is first displayed on the screen lower half side. And in order to display the portrait image which serves as a new candidate from this condition, actuation of changing a selection member is performed. The right key of the cross-joint key 109 or left key arranged in an equipment panel side is specifically operated, and a candidate is replaced. That is, when a right key is operated, a right end portrait image moves to a viewing-area B side, it becomes non-display, and the portrait image of a center and a left end shifts to right-hand side one by one according to this. Thereby, an opening is made at a left end, and a new candidate's portrait image is moved and displayed on this location from a viewing-area B side. In addition, when a left key is operated, contrary to ****, the portrait image of a center and a right end acts as a left shift, respectively, and a new candidate is displayed on a right end location. Therefore, if it continues operating either of the right-and-left keys, it will left-shift-go round or right shift go round, and renewal of sequential of a candidate's portrait image will be carried out.

[0092] If modification of a selection member is made as mentioned above, it will judge whether actuation which CPU1 progresses to step SJ7 shown in drawing 34 , and makes a member decision was performed. Here, if the depression of the enter key 111 is carried out that a predetermined member (portrait image) should be determined, a decision result will serve as "YES" and processing will be advanced to the following step SJ8. At step SJ8, the display-position coordinate (X, Y) of the object OBJA code of the member who made a current decision, and the display-position coordinate (X, Y) of the object OBJA code of the member for modification are changed mutually. Thereby, the portrait image of the member who made a current decision is fixed to a viewing-area A side.

[0093] Subsequently, if it progresses to step SJ9, it judges whether the escape key 110 which restores a

display condition was operated, and when the key 110 concerned is not operated, a decision result will serve as "NO", and this routine will once be completed. On the other hand, when an escape key 110 is operated in order to restore a display condition, a decision result serves as "YES" and processing is advanced to the following step SJ10. At step SJ10, the display-position coordinate (X, Y) of each object OBJA code of the portrait image currently displayed as a selection candidate is changed so that it may move to the viewing area B which becomes the outside of a screen display rectangle. And at the following step SJ11, the display-position coordinate (X, Y) of each object OBJA code of the selections display area evacuated to the viewing area B which becomes the outside of a screen display rectangle is changed so that it may display on the origin by the side of a viewing area A in a location. That is, at the above-mentioned steps SJ10-SJ11, if escape key actuation of restoring a display condition is made, while returning the portrait image currently displayed as a selection candidate to the viewing area B which becomes the outside of a screen display rectangle, each object of selections display area which evacuated to the viewing-area B side is moved to the original viewing-area A side, and it is set as a display condition. This changes from a band member modification screen to a band screen.

[0094] (g) If the object migration manipulation routine which an object color modification manipulation routine carries out **** of operation, and changes the display position of a portrait image according to modification of a band member is performed, CPU1 will perform the object color modification manipulation routine shown in drawing 35 through step SH2 (refer to drawing 32) mentioned above, and will advance processing to a step SK 1. It asks for the object which should cooperate to the processing of an object migration manipulation routine mentioned above, and should change a foreground color at a step SK 1. And according to the foreground color of the object set as the object of the color modification, the colour block number in object code is rewritten after a step SK 2. Hereafter, processing is explained according to the contents of color modification.

[0095] ** When changing into the Normal color, after completing band member modification actuation, in case it returns to a band screen, a color change of the object OBJA of a processing object is made at the Normal color showing a steady state. In this case, the decision result of a step SK 2 serves as "YES", it progresses to a step SK 3, and a colour block number with the Normal color of the target object OBJA is computed. And in the following step SK 4, the colour block number in the target object OBJA code is rewritten for the colour block number of the Normal color.

[0096] ** When changing into a selection color, in case a desired member is chosen at the time of band member modification, a color change of the object OBJA of a processing object is made at the selection color showing a selection condition. In this case, the decision result of a step SK 5 serves as "YES", it progresses to a step SK 6, and a colour block number with the selection color of the target object OBJA is computed. And in the following step SK 7, the colour block number in the target object OBJA code is rewritten for the colour block number of a selection color.

[0097] ** When changing into an active color, in case a desired member is determined at the time of band member modification, a color change of the object OBJA of a processing object is made at the active color showing a decision condition. In this case, the decision result of the step SK 8 shown in drawing 36 serves as "YES", it progresses to a step SK 9, and a colour block number with the active color of the target object OBJA is computed. And in the following step SK 7, the colour block number in the target object OBJA code is rewritten for the colour block number of an active color.

[0098] Thus, in an object color modification manipulation routine, it asks for the object which should cooperate to the processing of an object migration manipulation routine mentioned above, and should change a foreground color, and the colour block number in the corresponding OBJA code is rewritten according to the transition state of the object for which it asked to what shows either the "Normal color", a "selection color" and an "active color." A part for the frame part of a portrait image and the PERT name part arranged at the lower part of a portrait image are changing the foreground color as an object which follows, for example, makes a color change according to a transition state. Even if it is in equipment without a mouse by this, the so-called GUI environment can be realized and it becomes possible to improve operability.

[0099] (h) As it is beyond actuation of a genre selection screen manipulation routine, if a band screen

manipulation routine is completed, CPU1 will perform the genre selection screen manipulation routine shown in drawing 37 through the step SC 4 (refer to drawing 27) mentioned above, and will advance processing to a step SL 1. At a step SL 1, it judges whether it is under the genre selection mode in which the current mode chooses a performance genre, and when it is not that mode, a decision result here serves as "NO" and completes this routine. On the other hand, if it is under genre selection mode, a decision result will serve as "YES" and processing will be advanced to the following step SL 2. At a step SL 2, the generated event judges whether it is the event which requires image modification.

[0100] The event which requires image modification in a genre selection screen points out the actuation which chooses the character string for example, in the window WIN in a genre selection screen (refer to drawing 20) according to the vertical key stroke of the cross-joint key 109, and actuation of newly carrying out an alphabetic character input into this selected character string part, and changing a character string. And when above-mentioned selection actuation or character string modification actuation is made, the decision result of a step SL 2 serves as "YES", and processing is advanced to a step SL 3. In addition, a decision result here becomes being the actuation which does not need image modification with "NO", and this routine is completed.

[0101] When the event which requires image modification is generated and processing is advanced to a step SL 3, it judges whether CPU1 is that in which the actuation makes a character string change. Here, a decision result serves as "NO" at the case which chooses the character string in Window WIN according to the vertical key stroke of the cross-joint key 109, and processing is advanced to a step SL 4. At a step SL 4, the colour block modification processing (OBJB) routine (refer to drawing 31) which changes the foreground color of the alphabetic character (object OBJB) directly above the cursor location which moves in the inside of Window WIN according to actuation of the cross-joint key 109 is performed.

[0102] That is, as mentioned above, by the colour block modification processing (OBJB) routine, cursor is located in the "alphabetic character" part in the Normal condition (black), the colour block number of the object code which corresponds so that the "alphabetic character" may be displayed in an active color (pink) when display modification assignment is carried out is rewritten, and when it changes display modification assignment into the Normal condition from an active state, the colour block number of the Normal color is assigned contrary to this.

[0103] On the other hand, when inputting a new alphabetic character into the character string which chose the character string in Window WIN according to the vertical key stroke of the cross-joint key 109 and changing a character string, the decision result of the above-mentioned step SL 3 serves as "YES", and CPU1 advances processing to a step SL 5. At a step SL 5, the character string modification manipulation routine (refer to drawing 30) mentioned above is performed. It is based on the cursor location scrolled according to actuation of the cross-joint key 109 when making a character string change in a genre selection screen. From the storage area of ROM2 to namely, 18 character strings The OBJB code for eight lines is read according to the value of the pointer register i. When the read OBJB code is "a tooth space (null)", a transparence character (transparence color specification) is set up, when that is not right, it is set as the object B character (alphabetic character image) specified by the OBJB code, and this routine is completed.

[0104] (i) Actuation of a total control screen manipulation routine, next actuation of a total control screen manipulation routine are explained with reference to drawing 38 - drawing 40 . If an above-mentioned genre selection screen manipulation routine is completed, CPU1 will perform the total control screen manipulation routine shown in drawing 38 through a step SC 5 (refer to drawing 27), and will advance processing to a step SM 1. At a step SM 1, it judges whether the current mode is under the total control mode which sets up the mode of the whole equipment of operation, when it is not that mode, a decision result serves as "NO", and this routine is completed.

[0105] On the other hand, if it is under a total control mode, a decision result will serve as "YES" and processing will be advanced to the following step SM 2. At a step SM 2, the generated event judges whether it is the event which changes the image in a total control screen. Here, the event which changes an image points out the actuation which chooses the "carbon button" displayed on a total control screen

(refer to drawing 23) according to actuation of the cross-joint key 109, and depression actuation of the enter key 111 made in case the mode of operation matched with this selected "carbon button" is set up. And if the cross-joint key 109 or an enter key 111 is operated, the decision result of a step SM 2 will serve as "YES", and processing will be advanced to a step SM 3. In addition, a decision result here becomes being the actuation which does not need image modification with "NO", and this routine is completed.

[0106] Now, the cross-joint key 109 or an enter key 111 is operated, when the event which requires image modification occurs, CPU1 advances processing to a step SM 3, and it judges whether the actuation is what changes the display condition of a "carbon button." Here, when it is the event from which the display condition of a "carbon button" does not change, modification processing of others, such as initialization which a decision result serves as "NO", and advances processing to the following step SM 4, for example, initializes a setup of the whole equipment, is performed, and this routine is completed. On the other hand, when the event which changes the display condition of a "carbon button" occurs The decision result of the above-mentioned step SM 3 serves as "YES", and CPU1 advances processing to a step SM 5. The below-mentioned colour block modification manipulation routine for carbon buttons is performed. The display condition of an applicable carbon button The foreground color of the object which forms a "carbon button" is changed that it should carry out to either "the Normal condition which is visible to a concave" and the "selection condition" which were illustrated to drawing 24 , and "the active state which is visible to convex."

[0107] (j) If the event which changes the display condition of a "carbon button" occurs as the colour block modification manipulation routine for carbon buttons carried out **** of operation, CPU1 will perform the colour block modification manipulation routine for carbon buttons shown in drawing 39 through a step SM 5. In addition, in order to attain simplification of explanation, colour block modification about the "BeepOn carbon button" is mentioned as an example among various carbon buttons, and it explains, and goes by this routine. If this routine is performed, CPU1 will advance processing to step SN1, and it will judge whether a processing object is the "BeepOn carbon button." If it is except the "BeepOn carbon button", a decision result will serve as "NO" and will progress to step SN2. And at step SN2, colour block modification processing about the carbon button of others other than the "BeepOn carbon button" is performed. On the other hand, when a processing object is the "BeepOn carbon button", the decision result of step SN1 serves as "YES", the below-mentioned BeepOn carbon button colour block modification manipulation routine is performed, the foreground color of a "carbon button" formation object is changed according to the actuation gestalt made about the carbon button concerned, and a carbon button display mode is changed.

[0108] (k) If the book routine of a BeepOn carbon button colour block modification manipulation routine of operation is performed, CPU1 will advance processing to the step SP 1 shown in drawing 40 , and will set up the target foreground color according to the actuation gestalt made about the BeepOn carbon button. Hereafter, it explains according to an actuation gestalt.

** Carbon button selection is made in operating the cross-joint key 109 at the time of carbon button selection, and locating cursor in either the on-carbon button or the off-carbon button, and in this case, the target foreground color turns into a "selection color" showing a selection condition, and advances processing to a step SP 2. At a step SP 2, the colour block number corresponding to this "selection color" is computed, and it stores in Register destCblock. Since a "selection color" is colour block #8 in color look-up table CLT4OBJA shown in drawing 16 , "8" is set to Register destCblock.

[0109] ** Carbon button decision is made by carrying out ON actuation of the enter key 111 in the above-mentioned carbon button selection condition at the time of carbon button decision (setup), and in this case, the target foreground color turns into an "active color" showing a decision (setup) condition, at the step SP 3 which advances processing to a step SP 3, it computes the colour block number corresponding to this "active color", and stores it in Register destCblock. Since an "active color" is colour block #7 in color look-up table CLT4OBJA shown in drawing 16 , "7" is set to Register destCblock.

[0110] ** When ON actuation of the enter key 111 is carried out at the time of carbon button decision at

the time of setting cancellation, it becomes the "Normal color" which cancels a setup of the carbon button with which the actuation was made, and a pair of carbon button in exclusive relationship, and expresses a non-established state, and advance processing to a step SP 4. At a step SP 4, the colour block number corresponding to this the "Normal color" is computed, and it stores in Register destCblock. Since the "Normal color" is colour block #1 in color look-up table CLT4OBJA shown in drawing 16 , "1" is set to Register destCblock.
[0111] In this way, if the colour block number according to a display mode is set to Register destCblock, CPU1 will advance processing to a step SP 5. At a step SP 5, the object which should make a color change among object OBJA-A-OBJA-C (drawing 24 (**) reference) which constitutes the carbon button for modification (the on-carbon button or the off-carbon button) is extracted. Subsequently, at a step SP 6, the colour block value assigned to 4 bits of high orders of the OBJA code corresponding to the extracted object is rewritten for the colour block number in which it is stored by Register destCblock. Consequently, when made the Normal color, a display mode is changed so that it may be visible to convex, when a carbon button is made into an active color at a concave, and further, at the time of a selection color, a display mode is changed so that both in-between condition may be expressed.

[0112] (l) If the total control screen manipulation routine of a band save screen manipulation routine above-mentioned [of operation] is completed, CPU1 will perform the band save screen manipulation routine shown in drawing 41 through a step SC 6 (refer to drawing 27), and will advance processing to step SQ1. At step SQ1, it judges whether it is under the band save mode in which the current mode carries out band registration, when it is not that mode, a decision result serves as "NO", and this routine is completed.
[0113] On the other hand, if it is under band save mode, a decision result will serve as "YES" and processing will be advanced to the following step SQ2. At step SQ2, the generated event judges whether it is the event which changes the image in a band save screen. In addition, the event which changes an image points out the actuation which chooses the character string for example, in the window WIN displayed on a band save screen (refer to drawing 22) according to actuation of the cross-joint key 109, and the character string modification actuation which rewrites the selected character string. And when these actuation is made, the decision result of step SQ2 is set to "YES", processing is advanced to step SQ3 and the other actuation is performed, a decision result serves as "NO" noting that it is the actuation which does not need image modification, and this routine is completed.
[0114] Now, the cross-joint key 109 or an enter key 111 is operated, and suppose that the event which requires image modification occurred. If it does so, it will judge whether CPU1 is that in which advances processing to step SQ3 and the actuation makes "character string selection." Here, when "character string selection" is performed by the vertical key stroke of the cross-joint key 109, a decision result serves as "YES" and processing is advanced to step SQ4. At step SQ4, the foreground color of the object OBJA which performs the below-mentioned colour block modification processing (OBJA) routine, and is put on the same location as the selected character string (object OBJB) is changed into a "selection color", a selection condition is shown, and this routine is once completed.
[0115] On the other hand, when actuation other than "character string selection" is performed, the decision result of the above-mentioned step SQ3 serves as "NO", and processing is advanced to step SQ5. At step SQ5, when it judges whether actuation which rewrites the selected character string was performed and such actuation is not made, processing is advanced to step SQ6 by setting a decision result to "NO", and processing corresponding to the actuation is performed. On the other hand, when actuation which rewrites the selected character string is performed, the decision result of step SQ5 serves as "YES", and processing is advanced to step SQ7.
[0116] If it progresses to step SQ7, CPU1 will perform the character string modification manipulation routine (refer to drawing 30) mentioned above. It is based on the cursor location scrolled according to actuation of the cross-joint key 109 in this character string modification manipulation routine. From the storage area of ROM2 to 18 character strings The OBJB code for eight lines is read according to the value of the pointer register i. When the read OBJB code is "a tooth space (null)", a transparence

character (transparence color specification) is set up, and when that is not right, it is set as the object B character (alphabetic character image) specified by the OBJB code. And perform the colour block modification processing (OBJB) routine (refer to drawing 31) mentioned above, when an object object is the Normal color, an active color is made to make a color change, and when contrary to this, the Normal color is made to make a color change of CPU1, if it progresses to step SQ8.

[0117] (m) If the colour block modification processing (OBJA) routine shown in drawing 42 through the step SQ4 of a colour block modification processing (OBJA) routine above-mentioned [of operation] is performed, CPU1 will advance processing to a step SR 1. First, it asks for the object object which agrees with the cursor location moved according to actuation of the cross-joint key 109 at a step SR 1. And according to the foreground color of the object set as the object of the color modification, the colour block number in object code is rewritten after a step SR 2. Hereafter, processing is explained according to the contents of color modification.

[0118] ** When changing into the Normal color, a color change of the object OBJA from which it separated for selection by actuation of the cross-joint key 109 is made at the Normal color showing a steady state. In this case, the decision result of a step SR 2 serves as "YES", it progresses to a step SR 3, and a colour block number with the Normal color of the target object OBJA is computed. And in the following step SR 4, the colour block number in the target object OBJA code is rewritten for the colour block number of the Normal color.

[0119] ** When changing into a selection color, a color change of the object OBJA made applicable to selection by actuation of the cross-joint key 109 is made at the selection color showing a selection condition. In this case, the decision result of a step SR 5 serves as "YES", it progresses to a step SR 6, and a colour block number with the selection color of the target object OBJA is computed. And in the following step SR 7, the colour block number in the target object OBJA code is rewritten for the colour block number of a selection color.

[0120] Thus, while the foreground color of the object OBJA made applicable to selection by actuation of the cross-joint key 109 is changed into a "selection color" and a selection condition is shown, he is trying to return the object OBJA from which it separated for selection to the "Normal color" by the colour block modification processing (OBJA) routine.

[0121] (n) Actuation, now CPU1 of V blank interrupt manipulation routine are the process in which each routine explained until now is performed, and V blank interrupt manipulation routine which carries out the DMA transfer of object code required for a screen display, image data, and the color look-up table to the VDP5 and VRAM6 side synchronizing with the vertical-retrace-line period by the side of Display DP is performed. Whenever it enters at a vertical-retrace-line period, interruption activation is carried out, and this routine carries out the DMA transfer of the object code corresponding to screen mode, image data, and color look-up table at that time to the VDP5 and VRAM6 side, and, below, gives explanation of operation according to each screen mode.

[0122] ** initial-screen Mohd -- if interruption activation of this routine is first carried out whenever it enters at a vertical-retrace-line period, CPU1 will advance processing to the step SS 1 shown in drawing 43 . At a step SS 1, it judges whether current screen mode is initial-screen Mohd. And a decision result becomes being initial-screen Mohd with "YES", and processing is advanced to the following step SS 2. At a step SS 2, it judges whether the initial-screen transfer flag stored in Register IGF is "1." Since "1" is set when it changes to an initial screen in the screen change manipulation routine (refer to drawing 28) mentioned above, a decision result serves as "YES" and an initial-screen transfer flag advances processing to the following step SS 3, when changing to the initial screen.

[0123] At a step SS 3, the DMA transfer of background image data IM1BG and object A image data IM1OBJA (refer to drawing 7) which form an initial screen is carried out from ROM2 to VRAM6. Subsequently, if it progresses to a step SS 4, the DMA transfer of the above-mentioned image data IM1BG, color look-up table CLT1BG matched with IM1OBJA, and the CLT1OBJA will be carried out to the table area CLTA for object A of the color look-up table section 65 (refer to drawing 17) from ROM2. And it progresses to a step SS 5, and zero reset of the initial-screen transfer flag stored in Register IGF is carried out, the completion of a transfer is expressed, and processing is advanced to a

step SS 6.
[0124] At a step SS 6, OBJA code OBJ1OBJA for initial screens stored in RAM3 by the screen change manipulation routine mentioned above is read, and it transmits to the object memory 52 of the VDP5 interior. Consequently, VDP5 generates the initial screen which should be displayed with degree frame. In addition, when zero reset of the initial-screen transfer flag which a transfer is already completed and is stored in Register IGF is carried out when the decision result of the above-mentioned step SS 2 is "NO" namely, the above-mentioned step SS 6 is performed and OBJA code OBJ1OBJA for initial screens is transmitted to the object memory (OM) 52.
[0125] ** band screen mode -- in this case, the decision result of a step SS 7 serves as "YES", and advance processing to a step SS 8. At a step SS 8, it judges whether the band screen transfer flag stored in Register BGF is "1." Since "1" is set when it changes to a band screen in a screen change manipulation routine (refer to drawing 28), when changing to the band screen, a decision result serves as "YES", and a band screen transfer flag advances processing to the following step SS 9. At a step SS 9, the DMA transfer of background image data IM2BG which forms a band screen, and the object A image data IM2OBJA (refer to drawing 7) is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 10, the DMA transfer of the object B image data IM2OBJB which forms a band screen is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5 interior.
[0126] Subsequently, if it progresses to a step SS 11, CPU1 will carry out the DMA transfer of above-mentioned image data IM2BG, color look-up table CLT2BG respectively matched with IM2OBJA, and the CLT2OBJA to the table area CLTA for object A of the color look-up table section 65 (refer to drawing 17) from ROM2. Then, at a step SS 12, the DMA transfer of the color look-up table CLT2OBJB matched with image data IM2OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 from ROM2. And it progresses to a step SS 13 next, and zero reset of the band screen transfer flag stored in Register BGF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 14.
[0127] If it progresses to a step SS 14, CPU1 reads OBJA code OBJ2OBJA for band screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 15, will read OBJB code OBJ2OBJB for band screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the band screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the band screen transfer flag is carried out when the decision result of the above-mentioned step SS 8 is "NO" namely, the above-mentioned steps SS14 and SS15 are performed, and OBJA code OBJ2OBJA for band screens and OBJB code OBJ2OBJB for band screens are transmitted to the object memory (OM) 52, respectively.
[0128] ** genre selection screen mode -- the decision result of the step SS 16 shown in drawing 44 serves as "YES" in this case, and advance processing to a step SS 17. At a step SS 17, it judges whether the genre selection screen transfer flag stored in Register JSF is "1." Since this genre selection screen transfer flag is set to "1" when it is changed to a genre selection screen, a decision result here serves as "YES" and advances processing to the following step SS 18. At a step SS 18, the DMA transfer of background image data IM3BG and object A image data IM3OBJA (refer to drawing 7) which form a genre selection screen is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 19, and it forms a genre selection screen, the DMA transfer of the object B image data IM3OBJB is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5 interior.
[0129] Subsequently, if it progresses to a step SS 20, CPU1 will carry out the DMA transfer of the above-mentioned image data IM3BG, color look-up table CLT3BG respectively matched with IM3OBJA, and the CLT3OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. Then, at a step SS 21, the DMA transfer of the color look-up table CLT3OBJB matched with image data IM3OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 22 next, and zero reset of the band screen transfer flag stored in Register JSF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 23.

[0130] If it progresses to a step SS 23, CPU1 reads OBJA code OBJ3OBJA for genre selection screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 24, will read OBJB code OBJ3OBJB for genre selection screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the genre selection screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the genre selection screen transfer flag is carried out when the decision result of the above-mentioned step SS 17 is "NO" namely, the above-mentioned steps SS23 and SS24 are performed, and OBJA code OBJ3OBJA for genre selection screens and OBJB code OBJ3OBJB for genre selection screens are transmitted to the object memory (OM) 52.

[0131] ** total control screen mode -- in this case, the decision result of a step SS 25 serves as "YES", and advance processing to a step SS 26. At a step SS 26, it judges whether the total control screen transfer flag stored in Register TCF is "1." Since this flag is set to "1" when it is changed to a total control screen, a decision result here serves as "YES" and advances processing to the following step SS 27. At a step SS 27, the DMA transfer of background image data IM4BG and object A image data IM4OBJA (refer to drawing 7) which form a total control screen is carried out from ROM2 to VRAM6, respectively.

[0132] Subsequently, if it progresses to a step SS 28, CPU1 will carry out the DMA transfer of the above-mentioned image data IM4BG, color look-up table CLT4BG respectively matched with IM4OBJA, and the CLT4OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 29 next, and zero reset of the total control screen transfer flag stored in Register TCF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 30. At a step SS 30, OBJA code OBJ4OBJA for total control screens stored in RAM3 by the screen change manipulation routine mentioned above is read, and it transmits to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the total control screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the total control screen transfer flag is carried out when the decision result of the above-mentioned step SS 25 is "NO" namely, the above-mentioned step SS 30 is performed and OBJA code OBJ4OBJA for total control screens is transmitted to the object memory (OM) 52.

[0133] ** band save screen mode -- in this case, the decision result of a step SS 31 serves as "YES", and advance processing to a step SS 32. At a step SS 32, it judges whether the band save screen transfer flag stored in Register BSF is "1." Since this transfer flag is set to "1" when it is changed to a band save screen, a decision result here serves as "YES" and advances processing to the following step SS 33. At a step SS 33, the DMA transfer of background image data IM5BG and object A image data IM5OBJA (refer to drawing 7) which form a band save screen is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 34, and it forms a band save screen, the DMA transfer of the object B image data IM5OBJB is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5 interior.

[0134] Subsequently, if it progresses to a step SS 35, CPU1 will carry out the DMA transfer of the above-mentioned image data IM5BG, color look-up table CLT5BG respectively matched with IM5OBJA, and the CLT5OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. Then, at a step SS 36, the DMA transfer of the color look-up table CLT5OBJB matched with image data IM5OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 37 next, and zero reset of the band save screen transfer flag stored in Register BSF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 38.

[0135] If it progresses to a step SS 38, CPU1 reads OBJA code OBJ6OBJA for band save screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 39, will read OBJB code OBJ6OBJB for band save screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the band save screen which should be displayed with degree frame.

In addition, when a transfer is already completed and zero reset of the band save screen transfer flag is carried out when the decision result of the above-mentioned step SS 32 is "NO" namely, the above-mentioned steps SS38 and SS39 are performed, and OBJA code OBJ6OBJA for band save screens and OBJB code OBJ6OBJB for band save screens are transmitted to the object memory (OM) 52.
[0136] As explained above, while forming the alphabetic character displayed fixed on screens, such as a screen title, in this example by the object A image To the viewing area (display line) which displays the alphabetic character by which a display change is made on this object A image The object B image for several maximum alphabetic character minutes which can be displayed beforehand is arranged. When rewriting the alphabetic character, in case only the character name in the OBJB code corresponding to the display position (alphabetic character class) is changed and an alphabetic character is eliminated, transparence assignment of the colour block in the OBJB code is carried out, and it is changing into the non-display condition. Therefore, like the conventional image control unit, about both alphabetic character by which a display change is made on a screen, and alphabetic characters displayed fixed, such as a screen title, the class and display coordinate are managed separately, or the processing which moves the display position in corresponding object code out of a screen-display field becomes unnecessary in the case of character deletion. For this reason, if it becomes possible to aim at improvement in processing speed and this is put in another way, without using high-speed CPU, improvement in processing speed is attained, without inviting a cost rise.
[0137] If it is in the example mentioned above, it does not have pointing devices, such as a mouse. Moreover, ** The object A image which should change a foreground color according to actuation of the cross-joint key 109 or ENTA 111 is specified. Since the colour block number in the OBJA code corresponding to this was rewritten to what shows either the "Normal color", a "selection color" and an "active color" Improvement in operability, such as being able to realize the GUI environment which makes an image display mode adjustable corresponding to the contents of a key stroke, consequently preventing an operation mistake, is achieved. As this and a similar point, in the example mentioned above, the "carbon button" by which a screen display is carried out was constituted from object OBJA-A-OBJA-C (drawing 24 (**) reference), and in order to change the foreground color of these object OBJA-A-OBJA-C according to the contents of the actuation gestalt which makes an image change and to display the condition (the Normal condition, a selection condition, and active state) of relevance "a carbon button", the GUI environment excellent in visibility is attained.
[0138] Furthermore, the reversal flag X is formed in the OBJA code of the object A image displayed fixed on a screen in the example mentioned above. [whether the order of the write-in sequence at the time of storing object A image data in a line buffer according to the value of this flag X of the address is carried out, and] Or since it enabled it to specify whether it is made it and a reverse order, it is made to display as a usual image which wrote in one image data in order of the address, or it becomes possible to display the reverse image which wrote this in the reverse order and right and left of an image reversed. That is, the class of image which it can display since two images with a thing can be formed [which shall inverse-video-display one image data on usual] now can use image data efficiently for a system with little increase and memory space.
[0139] In addition, in this example, although reference was made about the case where it applies to the musical-sound control device by which the image control device by this invention is connected to a display, needless to say, the summary of this invention is not limited to this, for example, information machines and equipment, such as a personal computer and a word processor, can be applied also to personal digital assistant equipments (a pager, PHS, etc.) or video game equipment from the first. If it is equipment which realizes the user interface using an image in short, the summary of this invention can be applied and it excels in visibility by doing so, and the remarkable effectiveness of making the processing speed of a display control improve without inviting the cost rise of a product is acquired, realizing the operating environment which prevents an operation mistake.
[0140]
[Effect of the Invention] If the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled on a screen,

and an image arrangement means can display in this viewing area is arranged according to this invention If either of the modification images arranged in a viewing area is specified and the display mode is changed with a modification means, a modification image will be updated according to the changed display mode, and a screen display of this will be carried out according to arrangement within said viewing area by the display-control means. That is, since only the image with which a display mode is changed serves as a candidate for updating, it becomes possible to aim at improvement in processing speed, without using high-speed CPU, and if it puts in another way, processing speed can be improved, without inviting a cost rise.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention is used for example, for TV-game equipment etc., and relates to a suitable image control unit.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] In recent years, it connects with AV (audio-visual) terminal of a television receiver, and various utilization of the TV-game equipment used as a game toy, an intellectual training toy, etc. is carried out. This kind of equipment is equipped with the image control unit which generally consists of CPU, a ROM, RAM, a VRAM (Video RAM), etc. While transmitting at VRAM each image data of the static image memorized by ROM and a dynamic image to the bottom of directions of CPU By reading each image data transmitted to this VRAM, changing this into a video signal, and supplying AV terminal of a television receiver The animation of two or more characters which move according to game actuation is displayed, displaying a background image (static image) on the Braun tube.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] If the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled on a screen, and an image arrangement means can display in this viewing area is arranged according to this invention If either of the modification images arranged in a viewing area is specified and the display mode is changed with a modification means, a modification image will be updated according to the changed display mode, and a screen display of this will be carried out according to arrangement within said viewing area by the display-control means. That is, since only the image with which a display mode is changed serves as a candidate for updating, it becomes possible to aim at improvement in processing speed, without using high-speed CPU, and if it puts in another way, processing speed can be improved, without inviting a cost rise.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] By the way, in such an image control unit, although it is the translation which performs the same image control as a personal computer etc. fundamentally, it is necessary to make a product price possible the cheapest on the character in which it is used as a game toy or an intellectual training toy. For this reason, the hardware used inevitably will be restricted, for example, image memory (VRAM) capacity is stopped, and it is in the present condition of being hard to realize the operating environment called the so-called GUI (graphical user interface) environment, by some which omitted pointing devices, such as a mouse.
[0004] Moreover, in order to display a character string under the GUI environment, the character string displayed all over a window (display window) is treated as an object, and the character number and its display coordinate of each alphabetic character which should be displayed are set up separately. therefore, the character number and the display coordinate which correspond like a screen title also about the alphabetic character displayed fixed on a screen must be managed separately, and for actuation arrange the object of a character number which corresponds for erasing an alphabetic character on display further out of a viewing area is needed, the evil in_which disp of the case which displays a screen by these becomes slowly ****s [****] -- **. Although what is necessary is just to use high-speed CPU in order to raise processing speed, as mentioned above, a cost rise cannot be invited on product character, but this has been a technical technical problem.
[0005] Then, this invention was made in view of the situation mentioned above, and the main purpose is in offering the image control unit which realizes a high-speed screen display, without inviting a cost rise. Furthermore, as other purposes, image memory (VRAM) capacity is stopped and it is in offering the image control unit which can realize the false GUI environment also under the configuration which omitted pointing devices, such as a mouse.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] In order to attain the above-mentioned purpose, in invention according to claim 1 An image arrangement means to arrange the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled on the screen, and can be displayed in this viewing area, A modification means to specify either of the modification images arranged in said viewing area, and to change the display mode, It is characterized by providing a display-control means to direct a screen display for the modification image which updated and updated the modification image according to the display mode changed by this modification means according to arrangement within said viewing area.
[0007] Moreover, the 1st attribute data which specifies the class, display position, and foreground color of the fixed image displayed fixed on a screen in invention according to claim 2, An attribute data storage means to memorize the 2nd attribute data which specifies the class, display position, and foreground color of the modification image by which a display change is made on a screen, An image data storage means to memorize the image data matched with said 1st and 2nd attribute data, respectively, reading appearance of the image data which corresponds from said image data storage means according to said 1st and 2nd attribute data being carried out, respectively, and said modification image with an image display directions means to direct a screen display in piles, on said fixed image The rewriting means which rewrites said 2nd attribute data according to the directions when the directions which make a display change of said modification image are received, Only the modification image corresponding to said 2nd attribute data rewritten by this rewriting means is characterized by providing a display-control means to direct display modification.
[0008] According to invention according to claim 3, as a desirable embodiment subordinate to above-mentioned claim 2, said rewriting means is characterized by rewriting either the image class included in said 2nd attribute data, or a foreground color.
[0009] Moreover, in invention according to claim 4, said rewriting means is characterized by rewriting to transparence the foreground color contained in said 2nd attribute data, and changing into a non-display condition, when making a modification image into a null.
[0010] It has the flag which specifies whether the image data to which said 1st attribute data corresponds at invention according to claim 5 is supplied to said image display means in order of the address, or a reverse order is supplied, and is characterized by specifying either of the usual fixed image with which the image display means concerned was supplied in order of the address according to the value of the flag concerned, and the reverse image which the reverse order was supplied and the image reversed.
[0011] Furthermore, in invention according to claim 6, said fixed image is characterized by having been formed from two or more image data, having changed the foreground color of said 1st attribute data matched for each [these] image data of every, and making a display mode adjustable.
[0012] If the modification image for several minutes which defines beforehand the viewing area of the modification image whose modification of a display mode was enabled in this invention on a screen, and an image arrangement means can display in this viewing area is arranged If either of the modification images arranged in a viewing area is specified and the display mode is changed with a modification

means, a modification image will be updated according to the changed display mode, and a screen display of this will be carried out according to arrangement within said viewing area by the display-control means. That is, since only the image with which a display mode is changed serves as a candidate for updating, it becomes possible to aim at improvement in processing speed, without using high-speed CPU, and if it puts in another way, improvement in processing speed will be attained, without inviting a cost rise.

[0013]

[Embodiment of the Invention] It connects with a television receiver and the image control device by this invention displays a performance gestalt on the Braun tube, carries out the automatic performance of the performance information with the gestalt chosen from them, or may be applied to the musical-sound control device which generates the musical sound according to keyboard operation like the usual musical instrument. Below, such a musical-sound control unit is made into an example, and is explained with reference to a drawing.

[0014] A. See outside an example and explain the appearance of an example with reference to drawing 1 first. Drawing 1 shows the appearance of the musical-sound control unit 100 with which the image control unit by this invention was applied. In this drawing, 101 is the keyboard in which the key switch was arranged for every key, and generates the on-off signal according to the ****** actuation by the player. 102-111 are switches arranged in a panel side, respectively, and an electric power switch for 102 to turn on and off the power source of this equipment 100 and 103 are sound-volume switches which adjust the sound volume of generating musical sound. 104 is the Il Tempo switch which adjusts performance Il Tempo at the time of an automatic performance or automatic accompaniment. The tone switch with which 105 assigns a predetermined tone for every performance PERT, and 106 are rhythm switches which set the rhythm class of rhythm PERT at the time of an automatic performance or automatic accompaniment.

[0015] 107 is a stop switch operated in case an automatic performance or automatic accompaniment is stopped. 108 is a song switch operated in case the class of automatic performance data is chosen. 109 is a cross-joint key which consists of four-directions keys, and in case the cursor by which a screen display is carried out to the display terminal DP side mentioned later is moved to four directions, it is operated. 110 is an escape key, and in case it carries out setting cancellation and returns the time and a cursor location, it is operated. 111 is an enter key, and in case it carries out a definite input, it is operated. SP is a built-in loudspeaker arranged in the face of panel of the body of equipment. OUT1 and OUT2 are the audio output terminals and video outlet terminals which were prepared in the back-in-panels side side, respectively. The audio output and video outlet which are outputted from these output terminals OUT1 and OUT2 are connected to a television receiver or the well-known display terminal DP, and an image and voice (musical sound) are reproduced.

[0016] B. Describe the whole example configuration with reference to the configuration (1) of the whole configuration, next drawing 2 of an example. In addition, in this drawing, the same number is given to the element which is common into the part shown by the appearance of drawing 1 . In drawing 2 , 1 is CPU and controls the display and control section which consists of components 5, 6, and 7 which detect the event corresponding to panel switch actuation or ****** actuation, and mention it later according to the detected contents of an event, and the musical-sound control section which consists of components 8 and 9 based on a switch scan.

[0017] 2 is ROM the color look-up table which changes into color data (RGB data) the object code data showing attributes, such as a display coordinate location of the object image data for carrying out image display besides the various control programs performed in this CPU1 and object image data and a class, the background image data which forms a screen background, and these image data, or the song data for an automatic performance is remembered to be. The main data stored in this ROM2 are explained in full detail later on.

[0018] 3 is RAM, various register area is prepared as a work area of the above CPU 1, and the result of an operation, a flag value, etc. are stored temporarily. 4 is a keyboard and a switch interface circuitry. the on-off operation of the various actuation switches 103-111 with which this interface circuitry 4 is

arranged in a panel side (refer to drawing 1) -- or the switch event at the time of on-off operation of the key switch formed for every key of a keyboard 101 being carried out by ****** actuation is generated, and CPU1 is supplied.

[0019] 5 is a video display processor (it is hereafter described as VDP) which consists of various logical operation components, and functions as a well-known CRT controller. VDP5 is what bears the function which carries out a display control to the bottom of directions of CPU1. While carrying out the DMA transfer of the image data stored in ROM2 to the character generator memory (it mentions later) or below-mentioned VRAM6 inside self Display-control processing which extracts the image data which should be displayed [from] among the image data stored in each [these] memory, and defines that display gestalt and display position is performed, and the image data to which this processing was performed is changed and outputted to the RGB data showing a foreground color. In addition, this configuration of VDP5 is described later on.

[0020] 7 is an encoder, superimposes a perpendicular/Horizontal Synchronizing signal on the RGB data outputted from VDP5, and generates a composite video signal. The image the display control was carried out [the image] by VDP5 is displayed on CRT by supplying this composite video signal to a television receiver or the image input terminal of a display terminal DP. 8 is a sound source which consists of well-known wave memory read-out methods, reads the data point which corresponds from wave memory based on performance information, such as Keown / key-off when CPU1 occurs according to ****** actuation, or a velocity, or the automatic performance information (song data) read from ROM2 which CPU1 mentioned above, and generates musical-sound data. 9 is a D/A converter which changes and outputs the musical-sound data outputted from a sound source 8 to the audio output signal of analog format. Sound emission of the audio output signal outputted from this D/A converter 9 is carried out as a musical sound from the loudspeaker by the side of a display terminal DP through the output terminal OUT1 which sound emission was carried out from the built-in loudspeaker SP, or was mentioned above.

[0021] (2) Explain and go about the data configuration of ROM2, next the configuration of the main data stored in ROM2 with reference to drawing 3 - drawing 16 . Look-up table storage area ECLT the color look-up table which changes into color data (RGB data) the image data storage area EIMAGE where the background image data which forms the object image data and screen background corresponding to object storage area EOBJ the object code data which express with ROM2 the attribute, i.e., the object class, and its display coordinate location of the object which forms various screens are remembered to be, and these object codes is memorized, and each image data is remembered to be is prepared. Hereafter, the data configuration of these storage areas is described.

[0022] ** Explain the configuration of object storage area EOBJ with reference to the block diagram 3 of object storage area EOBJ - drawing 4 . In this storage area EOBJ, the object code for every various screens mentioned later assigns, and is memorized. Object code consists of two kinds of the objects B by which it is indicated by migration on a screen according to actuation of the object A displayed fixed on a screen or the above-mentioned cross-joint key 109, or an enter key 111. That is, only code OBJ1OBJA for object A and OBJ5OBJA which are displayed fixed on a screen, respectively are memorized by the object code OBJ1 for initial screens, and the object code OBJ5 for total control screens. In the object codes OBJ2, OBJ3, OBJ4, OBJ6, and OBJ7 corresponding to other screens, two kinds of codes of the object A displayed fixed on a screen and the object B by which it is indicated by migration are memorized.

[0023] Here, with reference to drawing 5 and drawing 6 , the data format of the object A code OBJA stored with the above-mentioned gestalt and the code OBJB for object B is explained. As for an OBJA/OBJB code, 1 word forms one object attribute by 3 words of WORD W0-W2 of 16 bit length. Magnitude (field), a display-position coordinate, a character name, a colour block (it mentions later) to be used of corresponding image data are expressed as an object attribute.

[0024] 9 bits (a bit 0 - bit 8) of low order of WORD W0 and W1 of the OBJA code / the OBJB code express the display-position coordinate on an object side (X, Y) so that it may illustrate to drawing 5 or drawing 6 . An object side is a dot flat surface defined by the coordinate field of - (0 0) (511,511). In the

object side OBJA (or OBJB), 336 dots wide which shares the above-mentioned zero (0 0), and a 224 dots (scan line) long field serve as the screen. In addition, the display-position coordinate (X, Y) expresses the location of the upper left corner in an object field. That object becomes non-display when the object location defined by this display-position coordinate (X, Y) is not contained in the field of the above-mentioned screen.

[0025] The reversal flag X showing whether corresponding image data is reversed is set to the bit 9 of WORD W1 in the OBJA code. It specifies whether this flag X carries out the order of the write-in sequence at the time of storing object image data in a line buffer of the address, or it is made it and a reverse order, the order of the address is specified at the time of "0", and a reverse order is specified at the time of "1." Therefore, influencing [of an image] becomes reverse [the usual image written in in order of the address, and the image written in the reverse order], and both images become mirror image relation. that is, it is possible to increase the class of image which can be displayed [which shall usually boil one image data and shall inverse-video-display it] in a system with little memory space by being able to form two images with a thing and doing in this way.

[0026] object area size -- a minimum of 8 -- it is formed by dot x8 dot and adjustable assignment of the size (X size, Y size) with the field in every direction in 4 bits (a bit 12 - bit 15) of high orders of WORD W0 and W1 may be carried out. For example, when X size is "0" and the breadth of an object is 8 dots and "1", it becomes the greatest 128-dot width of face at 16-dot width of face and the time of "15." In WORD W2, the colour block of the color look-up table section 65 which 4 bits (a bit 12 - bit 15) of the high order mention later is specified, the class of foreground color is decided, and the character name showing the character class of object is set to 12 bits of low order after a bit 12. In addition, a colour block means what blocked the color look-up table CLT mentioned later for every predetermined color number, and assigned the color code.

[0027] ** In the configuration image data storage area EIMAGE of the image data storage area EIMAGE, like storage area EOBJ mentioned above, as shown in drawing 7 , the image data for every various screens assigns, and is memorized. Image data is classified into three kinds such as the background image data which forms a screen background, the object A image data as which a class and a display coordinate location are specified in above-mentioned object A code, and the object B image data as which a class and a display coordinate location are specified in object B code. That is, either of the three above-mentioned kinds of image data is assigned, and the image data IM1-IM5 of each screen forms a screen as illustrated to drawing 7 . Three kinds of image data forms BG screen, an OBJA screen, and an OBJB screen, respectively, and these screens overlap and it forms the one display screen. In the case of this example, the highest priority is given to an OBJB screen, then it becomes the order of an OBJA screen and BG screen. Therefore, as sequence of lapping, an OBJB screen is arranged most in this side, and it is arranged in order of an OBJA screen and BG screen after that to a back side.

[0028] Object B image data is formed in the cel unit which consists of 8 dots of lengthwise directions, and 8 dots of longitudinal directions so that it may illustrate to drawing 8 . A 1-bit color code is assigned, and when it is "0", it becomes a transparence code at each dot which constitutes one cel. 1 word is expressed 4 words on account of WORD C0-C3 of 16 bit length as the color code for one cel is shown in drawing 9 . That is, the color code of the dot lines d0-d7 in 1 cel is stored in 8 bits of high orders of WORD C0, and the following dot lines d10-d17 are stored in 8 bits of low order of WORD C0. Henceforth, each dot lines d20-d27, d30-d37, --, d60-d67, and d70-d77 are similarly stored in WORD C1-C3 serially.

[0029] On the other hand, although object A image data is formed like the above-mentioned object B image data considering 8 dots of 8 dot x longitudinal directions of lengthwise directions as 1 cel unit so that it may illustrate to drawing 10 , a 4-bit color code is assigned to each dot in this case, and it serves as an image containing a transparent plane color carried out 16 color specification. 1 word is expressed 16 words on account of WORD D0-D15 of 16 bit length as the color code for one cel which forms Object OBJA is shown in drawing 11 . That is, the color code of the dots d00-d03 in 1 cel is stored in WORD D0, and the following dots d04-d07 are stored in WORD D1. Henceforth, each dots d10-d13, d14-d17, --, d74-d77 are similarly stored in WORD D1-D15 serially.

[0030] ** In configuration look-up table storage area ECLT of look-up table storage area ECLT, the color look-up table corresponding to the image data for every various screens is memorized as illustrated to drawing 12 . That is, the CLT data CLT1 for initial screens are divided into image data IM1BG for initial screens mentioned above, look-up table CLT1BG for BG screens which changes IM1OBJA into color data, respectively, and look-up table CLT1OBJA for OBJA screens, and such its a table gestalt is the same also in other CLT data CT2-CLT5 for screens.
[0031] Here, with reference to drawing 13 - drawing 16 , the example of representation of a color look-up table is explained. First, drawing 13 is drawing showing the table format of look-up table CLT2OBJA for OBJA screens in the CLT data CLT2 for band screens. This table CLT2OBJA is equipped with 16 kinds of look-up tables specified by colour block number #0-#15, and the color data D1-D15 of 15 classification by color are stored in each table. Therefore, color display of the image data used as the read-out address of this table may be carried out by a maximum of 16 color containing a transparent plane color.
[0032] Such a table is assigned for each [which forms the OBJA screen in a band screen] object OBJA of every. That is, it is specified with the "colour block value" (refer to drawing 5) stored in 4 bits of high orders in WORD W2 of the OBJA code mentioned above. In this table CLT2OBJA, the table which displays a band member's portrait image (it mentions later) on colour block number #0-#11 is assigned, and the title foreground color, the alphabetic character color, etc. are assigned to the colour block number #12-#15 remaining.
[0033] Next, drawing 14 is drawing showing the table format of look-up table CLT3OBJA for OBJA screens in the CLT data CLT3 for genre selection screens. As shown in this drawing, the object OBJA, i.e., a window, (display window) displayed fixed in the genre selection screen mentioned later, the window title, the scrolling foreground color, etc. are assigned as a colour block according to individual, respectively. On the other hand, table CLT3OBJB shown in drawing 15 is assigned to the object OBJB by which it is indicated by migration in a genre selection screen. In this case, since Object OBJB is the "alphabetic character" in which a 1-bit color code is assigned, "black" is assigned to colour block number #1 and "pink" is assigned to colour block number #2. In addition, it becomes transparence when colour block number #0 is assigned.
[0034] Drawing 16 is drawing showing the table format of look-up table CLT4OBJA for OBJA screens in the CLT data CLT4 for total control screens. As shown in this drawing, the colour block is assigned so that being "Normal", it being "a selection", and "it is active" may indicate the object OBJA which forms the "carbon button" displayed fixed in the total control screen mentioned later by the tri-state. In addition, "Normal", "a selection", and an "active" tri-state display are explained later on.
[0035] (3) Explain the configuration of VDP5 with reference to the configuration, next drawing 17 of VDP5. In this drawing, 50 is a CPU interface circuitry which is connected to a CPU bus, and delivers and receives CPU1 and data. 51 is a VRAM interface which manages data transfer with VRAM6. Background image data and object A image data are written in VRAM6 respectively through these interface circuitries 50 and 51 under directions of CPU1. Moreover, when reading these image data from VRAM6, like the time of writing, it reads from the CPU1 side through the above-mentioned interface circuitries 50 and 51, the address is given, and the image data read according to this is supplied to the below-mentioned controllers 59 and 63.
[0036] 52 is object memory in which the OBJA code for several object minutes displayed on a screen and the OBJB code are stored. The OBJA code and the OBJB code are data showing each attribute of the object A image data which forms Object OBJA, and the object B image data which forms Object OBJB, as illustrated to drawing 5 and drawing 6 . 53 is a control register which determines the mode of operation of VDP5. A control register 53 is a register of all 16 bit length, and a display-control gestalt is set up according to the register value set to each bit position by CPU1.
[0037] 54 is character generator memory and the object B image data which forms Object OBJB is memorized. The image data stored in this memory 54 forms an "alphabetic character", and as shown in drawing 8 , it is formed in the cel unit which consists of 8 dots of lengthwise directions, and 8 dots of longitudinal directions. A 1-bit color code is assigned, and when it is "0", it becomes a transparence

code at each dot which constitutes one cel. Although the object A image data which forms the object OBJA stored in VRAM6 mentioned above on the other hand is formed like the above-mentioned object B image data considering 8 dots of 8 dot x longitudinal directions of lengthwise directions as 1 cel unit as shown in drawing 10 , a 4-bit color code is assigned to each dot in this case, and it serves as an image containing a transparent plane color in which 16 color specification is possible.
[0038] 55 is a DMA controller which carries out DMA transfer control according to directions of CPU1. DMA controller 55 carries out block transfer of either the data for a transfer, i.e., background image data, Object A, B image data and an OBJA/OBJB code to either VRAM6 and the object memory 52 used as a transfer assignment place, or the character generator memory 54 from ROM2 mentioned above according to the transfer directions from CPU1. 56 is an object lead controller, it is read from the object memory 52 which mentioned above the object attribute corresponding to the horizontal scanning Rhine location by the side of a display, sorts the read object attribute in the sequence according to the quota gestalt of the object class specified by a control register 53, and stores it in the stack B memory 58 and the stack A memory 57 which mention the result later.
[0039] That is, by this controller 56, the OBJA/OBJB code of the Y coordinate which agrees in that scan line location whenever horizontal scanning Rhine is updated, and Y size is extracted from from among the OBJA code / the OBJB code stored in the object memory 52. And it asks for the display order on 1 horizontal-scanning Rhine based on the X coordinate of an OBJA/OBJB code and X size which were extracted, and the display priority specified by a control register 53. Next, in the sequence searched for, the OBJA code is stored in the stack A memory 57, and the OBJB code is stored in the stack B memory 58, respectively.
[0040] Therefore, by the stack A memory 57 and the stack B memory 58, the OBJA code which the object attribute which may be displayed in 1 horizontal-scanning period, respectively, i.e., an OBJA/OBJB code, will be memorized, and is memorized by these memory 57 and 58, and the OBJB code are read in order of a display by the controllers 59 and 60 mentioned later. The line buffer A controller 59 reads in order the OBJA code memorized in order of the display in the above-mentioned stack A memory 57, and reads the object A image data corresponding to the "character name" contained in the OBJA code concerned from VRAM6.
[0041] Moreover, this controller 59 writes the object A image data which read the X coordinate value included in the OBJA code from VRAM6 as the write-in address in a line buffer A61 (it mentions later). In case object A image data is written in a line buffer A, when the reversal flag X located in the bit 9 of WORD W1 in the OBJA code is "1", the write-in address is made into a reverse order, and when the reversal flag X is "0", on the other hand, it writes in in order of the address. Thereby, it may have comes to generate two mirror image-related images which right and left of an image reverse. Moreover, the colour block value (refer to drawing 3) extracted from the OBJA code is given to the object A image data written in a line buffer A61.
[0042] The line buffer B controller 60 reads in order the OBJB code memorized in order of the display in the stack B memory 58, reads the object B image data corresponding to the "character name" contained in the OBJB code concerned from the character generator memory 54, and writes it in a line buffer B62 (it mentions later). In the case of this writing, the X coordinate value in the OBJB code is used as the write-in address. Moreover, the "colour block" (refer to drawing 4) extracted from the OBJB code is given to the object B image data written in a line buffer B62.
[0043] The line buffer memory which stores temporarily the image data for 1 horizontal-scanning Rhine, respectively is prepared in the line buffer A61 and the line buffer B62 by two or more lines, and it is constituted so that the image data for the scan line displayed at least synchronizing with the horizontal scanning of a degree besides the image data for the scan line displayed synchronizing with a current horizontal scanning may also be stored temporarily. Moreover, it is controlled by the line buffer A61 and the line buffer B62 by the controllers 59 and 60 mentioned above so that buffer memory might be switched by turns and it might Rhine-read or Rhine write in synchronizing with a horizontal scanning. That is, apparent processing speed is improved by writing character data in the buffer memory of another side, reading character data from one buffer memory.

[0044] 63 is the BG code lead controller, reads the background image data stored in VRAM6, and stores it in the latter BG data register 64. The background image data stored in this register 64 serves as a static image which forms a screen background. While 65 gives screen priority, it is the color look-up table section which changes and outputs the image data of each screen where priority was given to color data (RGB data). Priority is given to each image data (Object A, B image data, and background image data) first outputted synchronizing with a horizontal scanning from the line buffer A61, the line buffer B62, and the BG data register 64 which were mentioned above in this table section 65. The priority showing the stacking-order foreword of a screen becomes settled according to the register value of a control register 53 mentioned above.
[0045] In the table section 65, it is divided into storage area CLTA which memorizes the color look-up table data corresponding to an object A image, and storage area CLTB which memorizes the color look-up table data corresponding to an object B image, and the DMA transfer of the color look-up table data of the class specified with the "colour block value" included in the object code corresponding to each image data is carried out to each [these] storage area. And this table section 65 chooses the color look-up table of a colour block number which corresponds according to the "colour block value" in the object code corresponding to each image data to which screen priority was given, and changes and outputs each image data to color data (RGB data) based on the selected table.
[0046] 66 is a D/A converter, and changes and outputs the color data (RGB data) outputted from the above-mentioned color look-up table section 65 to the chrominance signal for every color RGB. In addition, each chrominance signal is superimposed on a horizontal/Vertical Synchronizing signal in the encoder 7 mentioned above, and turns into a composite video signal. 67 is the synchronizing signal generating section which carries out the multiplying oscillation of the original ** clock of crystal oscillator X'tal, generates a horizontal synchronization / vertical-synchronization clock, and supplies this to each part in VDP5.
[0047] (4) Explain a screen configuration, next the configuration of the various screens which VDP5 forms based on the various data stored in ROM2. As mentioned above, to ROM2 An "initial screen", a "band screen", a "band member modification screen", Although it is the translation in which the object code, the image data, and look-up table data for forming a "genre selection screen", a "total control screen", and a "band save screen" are stored Here, the configuration of a "band screen", a "band member modification screen", a "genre selection screen", and a "total control screen" is described as a typical thing among these.
[0048] ** The configuration band screen of a band screen is located in this viewing-area [which is displayed on a display] A, and viewing-area A bottom, and consists of viewing areas B which serve as invisibility, without being displayed on a display. In addition, drawing 18 illustrates one mode of a viewing area A. In this viewing area A, parts which do not carry out display change, such as a title bar, and a title alphabetic character, a selection carbon button or a musician's portrait image, are formed by the object A image (OBJA), and, on the other hand, the character string CHR1 which carries out display change on this object A image according to actuation of the cross-joint key 109 or an enter key 111 - the CHR8 grade are formed by the object B image (OBJB).
[0049] In a viewing area B, when it is evacuated as an object A image and display selection of the selection carbon button, title bar, and portrait image by which display use is not carried out in the viewing area A is made according to selection actuation of the cross-joint key 109, the display coordinate location in the OBJA code is rewritten, and it moves to a viewing-area A side, and displays. moreover, color look-up table CLT2OBJA (refer to drawing 13) which mentioned above each object A image put on a viewing-area A side -- each -- it is matched with colour block #0-#15, and is expressed as the color data in an assignment colour block.
[0050] ** The configuration, next drawing 19 of a band member modification screen are drawing showing the band member modification screen displayed when it is operated so that a member change may be made in an above-mentioned band screen. While moving the selections display area displayed on the lower half of a band screen to a viewing area B, the portrait image for 3 persons is instead chosen and expressed as a band member modification screen from from among the portrait images for two or

more person minutes of several arranged at the viewing area B. Drawing 19 illustrates such a condition. By the way, the portrait images P1 and P2 shown in drawing 19 are formed of the same object A image data, and two mirror image-related images with which right and left of an image reverse the reversal flag X in the OBJA code mentioned above when generating the image of one of these by setting to "1" have been obtained. In addition, also in such a band member modification screen, the part which does not carry out display change is formed by the object A image (OBJA), and the character string which carries out display change is formed by the object B image (OBJB).

[0051] ** The window WIN used in case a performance genre is chosen as middle of the screen, as shown in drawing 20 is expressed as the configuration genre selection screen of a genre selection screen. This window WIN consists of object OBJA-50 which display object OBJA-1 - OBJA8, and the condition that can be scrolled which constitutes a title alphabetic character, a title bar (object OBJA-52) by which OBJA-51 are arranged, and a color adjustable carbon button (object OBJA-51-OBJA-60) of eight lines so that it may illustrate to drawing 21 .

[0052] With each carbon button (object OBJA-51-OBJA-60), 18 trains of objects OBJB of the 8 dot x 8-dot size of ends of a road are arranged. That is, matrix arrangement of object OBJB-128-OBJB-271 is carried out. These object OBJB-128-OBJB-271 form an alphabetic character, or the rarefaction is carried out, and they are set as display ranking higher than the above-mentioned object OBJA-51-OBJA-60. Therefore, it seems that the character string is drawn on the carbon button. Moreover, each carbon button may change a foreground color, as it is shown by changing the colour block number of color look-up table CLT3OBJA (refer to drawing 14) mentioned above according to the vertical key stroke of the cross-joint key 109 whether it is in a selection condition.

[0053] ** Also in the band save screen shown in the block diagram 22 of a band save screen, the same window WIN as **** is arranged. In the window WIN in a van save screen, the class of object OBJB which forms the alphabetic character on each carbon button is changed, and a character string can be rewritten now. In that case, the foreground color of Object OBJB changes the colour block number of color look-up table CLT3OBJB shown in drawing 15 to #2 (pink) from #1 (black), rewrites it, and can indicate that it is inside.

[0054] ** The configuration, next drawing 23 of a total control screen are drawing showing an example of a total control screen. On this screen, the "carbon button" which sets up the terms and conditions of equipment 100 of operation is formed of Object A. Each carbon button is constituted so that it may be in the tri-state of the active state (c) which is visible to convex [which seem to illustrate to the drawing 24 (**) in a concave / the Normal condition (a), the selection condition (b), and convex]. That is, by OBJA-D and E which form the alphabetic character on OBJA-A which forms the left brink upper limb of a carbon button, OBJ-B which forms the right-margin-of-heart margo inferior of a carbon button, OBJA-C which forms a carbon button center section, and a carbon button, a carbon button is formed, the foreground color of these object OBJA-A-OBJA-C is changed according to the contents of setting actuation, and the condition of above-mentioned (a) - (b) is expressed as shown in the drawing 24 (**).

[0055] For example, "RESUME" in drawing is explained as an example. First, with the carbon button of a non-established state, the color data D13, D14, and D15 in colour block number #0 of color look-up table CLT4OBJA are assigned to above-mentioned object OBJA-A-OBJA-C, respectively, and it changes into the Normal condition (a) which is visible to a concave. When carbon button selection is made by the right-and-left key stroke of the cross-joint key 109 from this condition, the color data D13, D14, and D15 of colour block number #8 are assigned to object OBJA-A-OBJA-C, a foreground color is changed, and a selection condition (b) is expressed. And if an enter key 111 is pushed and a setup is made to decide, the color data D13, D14, and D15 of colour block number #7 will be assigned to object OBJA-A-OBJA-C, and it will be made the active state (c) which is visible to convex. It becomes possible to realize a GUI operating environment to which is not equipped with pointing devices, such as a mouse, but ** also clicked the carbon button by doing in this way.

[0056] C. Explain actuation of an example, next actuation of the example by the above-mentioned configuration with reference to drawing 25 - drawing 43 . After describing the outline of whole actuation first, about the detail of the image processing in connection with the summary of this

invention, sequential explanation is given and it goes by below.
[0057] (1) An injection of the electric power switch 102 of the equipment 100 which is whole (actuation a) outline flow this example operates CPU1 according to the flow shown in drawing 25 . That is, if an electric power switch 102 is switched on, it progresses to a step SA 1, and it will progress to a step SA 2, after carrying out zero reset of the register and flag which are formed in each part of RAM3, VRAM6, VDP5, and a sound source 8 or initializing setting initial value etc. At a step SA 2, while scanning a keyboard 101 and a panel switch group, detecting a ****** actuation event or a switch actuation event, generating musical sound or performing pronunciation processing / music processing which sets the various performance parameters at the time of carrying out musical-sound generating according to the detected event, the image processing for generating a screen display corresponding to these processings is performed.
[0058] Subsequently, if a step SA 2 is completed, CPU1 will advance processing to a step SA 3, and it will judge whether an auto-power-off setup was carried out. Here, if it is not in a power-off condition, a decision result serves as "NO", processing is returned to a step SA 2, the Maine flow processing is continued, on the other hand, when a power-off setup is carried out, a decision result will serve as "YES" and processing will be advanced to the following step SA 4. At a step SA 4, after, performing resume processing which carries out memory evacuation of each part established state in equipment 100, silence processing which carries out mute of the musical sound under pronunciation for example, power-off processing which carries out power-source OFF is performed, and a halt of operation is carried out.
[0059] (b) If processing of the Maine flow CPU 1 progresses to the above-mentioned step SA 2, the Maine flow shown in drawing 26 will be performed, and it will progress to a step SB 1. At a step SB 1, a keyboard scan is performed that ****** actuation of a keyboard 101 should be detected. Then, at a step SB 2, while generating performance information, such as Keown, key-off, a keycode, and a velocity, based on the ****** actuation event detected in this keyboard scan, pronunciation processing which directs musical-sound generating in a sound source 8 according to this performance information is performed. Subsequently, at a step SB 3, in order to detect switch actuation of panel switches 103-111, a panel switch scan is performed and a switch actuation event is generated.
[0060] Next, at a step SB 4, the switch actuation which creates the automatic performance information on time series among the generated switch actuation events is extracted, and an event matrix is created. It becomes the performance pattern which will serve as this event matrix from a pitch and pronunciation timing when becoming the rhythm pattern which enumerated pronunciation timing and creating automatic performance information if automatic rhythm performance information is created. Subsequently, if it progresses to a step SB 5, music processing which sets up the effectiveness gestalt given when CPU1 generates the rhythm pattern or performance pattern expressed as the above-mentioned event matrix as a musical sound will be performed. Then, that the screen corresponding to the image processing in connection with a summary, i.e., the above-mentioned switch actuation event, of this invention should be generated, a display control is carried out, and at the continuing step SB 7, CPU1 performs that alien-system processing, and completes this Maine flow.
[0061] (2) Explain actuation of an image processing, next the contents of the image processing performed in the above-mentioned step SB 6 in full detail. If processing of CPU1 progresses to the above-mentioned step SB 6, the image-processing routine shown in drawing 27 will be started, and processing will be advanced to a step SC 1. At a step SC 1, it judges whether the switch actuation event which changes whether there is a Mohd change and a screen display that is, occurred. Here, if there is a switch actuation event which changes a screen display, a decision result serves as "YES", processing is advanced to the following step SC 2, and when that is not right, processing will be advanced to the below-mentioned step SC 3. If it progresses to a step SC 2, CPU1 will perform screen change processing which extracts the object code which corresponds from ROM2 according to the contents of a screen which should be changed, and is stored in the object memory 52 of the VDP5 interior. In addition, about the detail of screen change processing, it mentions later.
[0062] Subsequently, at steps SC3-SC6, the display control corresponding to the switch actuation event generated under each screen mode is performed. That is, if it is in a step SC 3, the display mode of the

character strings CHR1-CHR8 displayed all over a band screen (refer to drawing 18) according to actuation of the cross-joint key 109 mentioned above, or an escape key 110 or an enter key 111 is changed. Next, at a step SC 4, the character string in the window WIN displayed on a genre selection screen (refer to drawing 20) according to actuation of the cross-joint key 109 or an enter key 111 is indicated by migration, or the display control which makes a color change is performed. And at a step SC 5, the display control which changes the foreground color of the "carbon button" for every item in a total control screen (refer to drawing 23) according to switch actuation is performed. Furthermore, on a band save screen (refer to drawing 22), according to actuation of the cross-joint key 109 or an enter key 111, the character string in Window WIN is indicated by migration, or the display control which makes a color change is performed in a step SC 6.

[0063] About the detail of the display control made in these steps SC3-SC6, it mentions later. Moreover, CPU1 which performs the display control by these steps SC3-SC6 carries out interrupt processing of the V blank interrupt manipulation routine (it mentions later) for every fixed period, and updates the screen by which the display control was carried out. That is, in the process in which steps SC3-SC6 are performed, CPU1 performs V blank interrupt manipulation routine synchronizing with the vertical-retrace-line period by the side of Display DP, and carries out the DMA transfer of object code required for a screen display, image data, and the color look-up table to the VDP5 and VRAM6 side. And in VDP5, a screen is updated based on the various above-mentioned data by which the DMA transfer was carried out in this way to the character generator memory 54 inside VRAM6 or self, and the object memory 52.

[0064] (a) Explain actuation of a screen change manipulation routine, next actuation of a screen change manipulation routine for every screen.

** If the switch actuation event to an initial screen which changes a screen display as changed and mentioned above occurs, CPU1 will perform the screen change manipulation routine shown in drawing 28 through a step SC 2, and will advance processing to a step SD 1. At a step SD 1, the generated switch actuation event judges whether it is what directs the change to an initial screen. Here, if it is the event which directs the change to an initial screen, the decision result of a step SD 1 will serve as "YES", and, as for CPU1, processing will be advanced to a step SD 2. At a step SD 2, the value of the initial-screen transfer flag stored in Register IGF is set to "1."

[0065] Subsequently, if it progresses to a step SD 3, the object code OBJ1 (refer to drawing 3) for initial screens stored in ROM2 will be stored in the work area of RAM3. By doing in this way, during a vertical blanking interval, the background image data, the object A image data, and the CLT data CLT1 for initial screens which form an initial screen are transmitted to the VRAM6 and VDP5 side by V blank interrupt manipulation routine mentioned later, and, thereby, an updating indication of the initial screen is given.

[0066] ** When the switch actuation event which directs a change on the change band screen to a band screen occurs, the decision result of the above-mentioned step SD 1 serves as "NO", advance processing to a step SD 4, and judge whether it is a change on a band screen. And the decision result of a step SD 4 serves as "YES", and advances processing to the following step SD 5 because a change on a band screen is performed in this case. At a step SD 5, the value of the band screen transfer flag stored in Register BGF is set to "1." Subsequently, if it progresses to a step SD 6, code OBJ2OBJA for object A will be extracted from from among the object codes OBJ2 (refer to drawing 3) for band screens stored in ROM2, and it will store in the work area of RAM3.

[0067] Then, if it progresses to a step SD 7, CPU1 will extract code OBJ2OBJB for object B from from like the above-mentioned step SD 6 among the object codes OBJ2 (refer to drawing 3) for band screens stored in ROM2, and will store it in the work area of RAM3. And in the below-mentioned V blank interrupt manipulation routine, image data IM2 (refer to drawing 7) for band screens and the CLT data CLT2 (refer to drawing 12) for band screens which form a band screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side by this, and a band screen is formed.

[0068] ** When the switch actuation event which directs a change on a genre selection screen, next a change on a genre selection screen occurs, all serve as "NO", and the decision result of the above-

mentioned steps SD1 and SD4 advances processing to a step SD 8, and judges whether it is a change on a genre selection screen. And the decision result of a step SD 8 serves as "YES", and processing is advanced to the following step SD 9 because a change on a genre selection screen is performed in this case. At a step SD 9, the value of the genre selection screen transfer flag stored in Register JSF is set to "1." Subsequently, if it progresses to a step SD 10, code OBJ4OBJA for object A will be extracted from from among the object codes OBJ4 (refer to drawing 3) for genre selection screens stored in ROM2, and it will store in the work area of RAM3.

[0069] Then, if it progresses to a step SD 11, CPU1 will extract code OBJ4OBJB for object B from from like the above-mentioned step SD 10 among the object codes OBJ4 (refer to drawing 3) for genre selection screens stored in ROM2, and will store it in the work area of RAM3. Consequently, based on the below-mentioned V blank interrupt manipulation routine, image data IM3 (refer to drawing 7) and the CLT data CLT3 (refer to drawing 12) for genre selection screens which form a genre selection screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side, and a genre selection screen is updated.

[0070] ** If the switch actuation event which directs a change on the change total control screen to a total control screen occurs, all will serve as "NO", and the decision result of the above-mentioned steps SD1, SD4, and SD8 will advance processing to a step SD 12, and will judge whether it is a change on a total control screen. And since a change on a total control screen is performed in this case, the decision result of a step SD 12 serves as "YES", and advances processing to the following step SD 13.

[0071] At a step SD 13, the value of the total control screen transfer flag stored in Register TCF is set to "1." Subsequently, if it progresses to a step SD 14, the object code OBJ5 (refer to drawing 3) for total control screens stored in ROM2 will be stored in the work area of RAM3. Consequently, a total control screen is updated by transmitting image data IM4 (referring to drawing 7) and the CLT data CLT4 (referring to drawing 12) for total control screens which form a total control screen during a vertical blanking interval to the VRAM6 and VDP5 side based on the below-mentioned V blank interrupt manipulation routine.

[0072] ** If the switch actuation event which directs a change on the change band save screen to a band save screen occurs, all will serve as "NO", and the decision result of steps SD1, SD4, SD8, and SD12 mentioned above will advance processing to a step SD 15, and will judge whether it is a change on a band save screen. And the decision result of a step SD 15 serves as "YES", and processing is advanced to the following step SD 16 because a change on a band save screen is performed in this case. At a step SD 16, the value of the band save screen transfer flag stored in Register BSF is set to "1." Subsequently, if it progresses to a step SD 17, code OBJ6OBJA for object A will be extracted from from among the object codes OBJ6 (refer to drawing 3) for band save screens stored in ROM2, and it will store in the work area of RAM3.

[0073] Then, at a step SD 18, code OBJ6OBJB for object B is extracted from from like the above-mentioned step SD 17 among the object codes OBJ6 (refer to drawing 3) for band save screens stored in ROM2, and it stores in the work area of RAM3. Consequently, based on the below-mentioned V blank interrupt manipulation routine, image data IM5 (refer to drawing 7) and the CLT data CLT5 (refer to drawing 12) for band save screens which form a band save screen during a vertical blanking interval are transmitted to the VRAM6 and VDP5 side, and a band save screen is updated. In addition, when other different events from the event which directs a screen change occur, as for each decision result in the above-mentioned steps SD1, SD4, SD8, SD12, and SD15, all serve as "NO", and CPU1 completes this routine.

[0074] (b) When events other than a screen change occur when a band screen manipulation routine carries out **** of operation and a screen change manipulation routine is completed or, CPU1 performs the band screen manipulation routine shown in drawing 29 through the step SC 3 mentioned above, and advances processing to a step SE 1. At a step SE 1, it judges whether a screen current on display is a band screen. Here, in not being Mohd who displays a band screen, a decision result serves as "NO" and completes this routine. On the other hand, while displaying the band screen, a decision result serves as "YES" and processing is advanced to the following step SE 2. If it progresses to a step SE 2, as for

CPU1, the generated event will judge whether it is the event which requires image modification.
[0075] The event which requires image modification points out band member modification actuation of replacing the portrait image of the case where the character string CHR1 in a band screen (refer to drawing 18) is chosen according to the vertical key stroke of the cross-joint key 109, and the musician displayed for every PERT. And when actuation which needs image modification is performed, the decision result of the above-mentioned step SE 2 serves as "YES", and processing is advanced to a step SE 3. On the other hand, when the actuation which does not need image modification is made, a decision result serves as "NO" and completes this routine.
[0076] Now, when actuation which needs image modification is performed, CPU1 advances processing to a step SE 3, and it judges whether it is that by which the generated event changes a character string. Here, if the upper key of the cross-joint key 109 or a bottom key is operated that either of the character strings CHR1 in a band screen (refer to drawing 18) should be chosen and a cursor location is scrolled, a decision result will serve as "YES" and processing will be advanced to the following step SE 4. At a step SE 4, the character string modification manipulation routine (it mentions later) which chooses either of the character strings CHR1 according to this scrolling actuation is performed. Subsequently, at a step SE 5, the colour block number matched with the selected character string (object OBJB) is rewritten, and this routine is completed, after performing the colour block modification manipulation routine (it mentions later) which changes the foreground color of that character string.
[0077] On the other hand, when actuation other than character string modification is made when a decision result is "NO" in the above-mentioned step SE 3 namely, processing is advanced to a step SE 6. At a step SE 6, it judges whether the actuation in which it succeeded is actuation, i.e., band member modification actuation of replacing with other things either of a musician's portrait images displayed all over a band screen, of changing a band member. And if the actuation is performed, a decision result here will serve as "YES" and processing will be advanced to the following step SE 7. At a step SE 7, while displaying PERT's portrait image by which modification assignment was carried out on an assignment coordinate location, the band member modification manipulation routine (it mentions later) which changes the foreground color of the "PERT name" part arranged at the lower part of the portrait image made applicable to modification is performed, and this routine is completed.
[0078] (c) If the upper key of the cross-joint key 109 or a bottom key is operated that either of the character strings CHR1 in a band screen should be chosen and a cursor location is scrolled as the character string modification manipulation routine carried out **** of operation, CPU1 will perform the character string modification manipulation routine shown in drawing 30 through a step SE 4, and will advance processing to a step SF 1. The direction of operated scrolling is distinguished at a step SF 1. The scrolling direction progresses to a step SF 2, when the upper key of the cross-joint key 109 is operated a "top" that is, and the decrement of the direction address of Y which shows the display area on ROM2 is carried out by one line.
[0079] That is, code OBJ2OBJB for object B which forms the alphabetic character in which a migration display is possible all over a band screen is stored in the form corresponding to the display gestalt on a screen in object storage area EOBJ of ROM2. The storage area serves as direction of X (direction of train) 18 character string, and the direction (line writing direction) of six lines of Y, and a screen display of this is carried out as a character string CHR1. Therefore, when the upper key of the cross-joint key 109 is operated and it scrolls to the bottom on a screen, 1 decrement of the direction read-out address of Y of the OBJB code in the above-mentioned storage area is carried out. Moreover, when the bottom key of the cross-joint key 109 is operated and it scrolls to the bottom on a screen like this, based on distinction of the above-mentioned step SF 1, processing is advanced to a step SF 3 and the direction read-out address of Y is incremented one time in this case.
[0080] Subsequently, if it progresses to a step SF 4, CPU1 will read the present address (X, Y) updated according to scrolling actuation, and will use it as an initiation pointer. And "1" is set to the pointer register i at the following step SF 5. In addition, a sequential increment is carried out, and the value of this pointer register i is read, and is treated as a relative address on the basis of an initiation pointer so that it may mention later. and criteria [cursor location / which was scrolled henceforth / a step SF 6] --

carrying out -- 18 character string [from the above-mentioned storage area] x -- the OBJB code for six lines is read according to the value of the pointer register i. That is, at a step SF 6, it judges whether 18 character strings and the OBJB code for six lines were read. Here, when 18 character strings and the OBJB code for six lines are read, a decision result serves as "YES" and progresses to a step SF 7. At a step SF 7, a read-out initiation pointer is reset and this routine is completed.
[0081] On the other hand, when read-out of the OBJB code is not completed, the decision result of the above-mentioned step SF 6 serves as "NO", it progresses to a step SF 8, and the OBJB code train which forms the character string specified based on the value and the present address (X, Y) of the pointer register i is read. Subsequently, at a step SF 9, the read OBJB code train is interpreted a single character every, and it judges whether the interpreted OBJB code is "a tooth space (null)." Here, a decision result becomes being "a tooth space (null)" with "YES", processing is advanced to the following step SF 10, and it is set as a transparence character (transparence color specification).
[0082] On the other hand, when it is not "a tooth space (null)", the decision result of the above-mentioned step SF 9 serves as "NO", processing is advanced to a step SF 11, and it is set as the object B character (alphabetic character image) specified by the OBJB code. And after this, CPU1 advances processing to a step SF 12, stores the interpreted OBJB code in the object memory 52 of the VDP5 interior, increments the pointer register i one time in the continuing step SF 13, and returns processing to the above-mentioned step SF 6.
[0083] It is based on the cursor location scrolled according to actuation of the cross-joint key 109 in character string modification processing. From the storage area of ROM2 to thus, 18 character strings The OBJB code for six lines is read according to the value of the pointer register i. When the read OBJB code is "a tooth space (null)", a transparence character (transparence color specification) is set up, and when that is not right, he is trying to set it as the object B character (alphabetic character image) specified by the OBJB code. That is, all the alphabetic characters by which a display change is made are set as the object B image, and since what is necessary is to specify the corresponding OBJB code and just to specify it as a transparent plane color in the case of an elimination display, in case a display change is made, compared with the conventional thing which specifies a display gestalt and a display coordinate location, it can grow into every character (alphabetic character) with a high-speed screen display.
[0084] (d) Explain actuation of a colour block modification processing (OBJB) routine, next actuation of the colour block modification manipulation routine which changes the foreground color of the object OBJB set as the object of display modification with reference to drawing 31 . An object B image can change a colour block number, and can change a foreground color to "black" and "pink" by turns since the 1 bit [per dot] color code is assigned as mentioned above, for example, so that color look-up table CLT3OBJB illustrated to drawing 15 may show. When changing the colour block of Object OBJB, "black" is equivalent to the below-mentioned "Normal color", and "pink" is equivalent to the below-mentioned "active color."
[0085] Suppose that modification of a character string (object OBJB) was completed by the now, for example, above-mentioned character string, modification manipulation routine. If it does so, CPU1 will perform the colour block modification processing (OBJB) routine shown in drawing 31 through the step SE 5 mentioned above in order to change the foreground color of this changed character string, and will advance processing to step SG1. At step SG1, the object first made applicable to modification is specified. Subsequently, at step SG2, it judges whether the foreground color currently assigned to the object made into the object is the "Normal color." And if it is the "Normal color", a decision result will serve as "YES", processing will be advanced to the following step SG3, and the colour block number which assigns an "active color" to the object will be computed.
[0086] On the other hand, when the foreground color currently assigned to the target object is an "active color", the decision result of the above-mentioned step SG2 serves as "NO", and CPU1 advances processing to step SG4, and computes the colour block number which assigns the "Normal color" to the object. And if it progresses to step SG5 after this, the colour block number in the target object OBJB code will be rewritten for the colour block number computed at the above-mentioned step SG3 or step

SG4. Thus, in colour block modification processing (OBJB), when making a display change in the state of Normal, the colour block number of an active color is assigned, and in [that] being opposite, it assigns the colour block number of the Normal color.
[0087] (e) In the band screen of a band member modification manipulation routine of operation, if band member modification actuation of replacing with other things either of five kinds of a musician's portrait images displayed all over a screen is made, CPU1 will perform the band member modification manipulation routine shown in drawing 32 through the above-mentioned step SE 7 (refer to drawing 29), and will progress to step SH1. An object migration manipulation routine (it mentions later) is performed, and the specified location is made to indicate a member's portrait image by migration at step SH1. And at step SH2, the foreground color of the object OBJA corresponding to this portrait image by which it was indicated by migration is changed based on actuation of an object color modification manipulation routine (it mentions later).
[0088] (f) If the above-mentioned step SH1 of an object migration manipulation routine of operation is performed, CPU1 will perform the object migration manipulation routine shown in drawing 33 , and will advance processing to step SJ1. At step SJ1, it judges whether the actuation which displays a selection member was made. The actuation which displays a selection member here moves cursor to the location of the character string CHR7 ("MEMBER") displayed on the lower part of the band screen illustrated to drawing 18 , and after pushing an enter key 111 there and making band member maintenance mode change, it points out actuation of specifying the "PERT" who should change. When such actuation is made, the decision result of the above-mentioned step SJ1 serves as "YES", and processing is advanced to the following step SJ2. on the other hand, the actuation which displays a selection member should do -- when there is nothing, it progresses to the below-mentioned step SJ4.
[0089] At step SJ2, the display-position coordinate (X, Y) of each object OBJA code which forms selections display area is changed so that it may move to the viewing area B which becomes the outside of a screen display rectangle. In addition, the selections display area said here points out the screen range of a lower half from the portrait image in a band screen. And in the following step SJ3, the display-position coordinate (X, Y) of the object OBJA code to the selected member is moved to a viewing-area A side as a substitute of the above-mentioned selections display area. In addition, the selected member means a selection candidate's portrait image beforehand assigned to the "PERT" set as the object of member modification. That is, in the above-mentioned step SJ2 - SJ3, if the actuation which displays a selection member is made, while moving selections display area to the viewing area B which becomes the outside of a screen display rectangle, the portrait image of the selection candidate stationed at the viewing-area B side is moved to a viewing-area A side as a substitute of selections display area. Thereby, the band member modification screen illustrated to drawing 19 is formed.
[0090] Next, if it progresses to step SJ4, CPU1 will judge whether actuation, i.e., the actuation on which the portrait image which serves as a new candidate is displayed, of changing a selection member was performed. And if actuation of changing a selection member is made, a decision result will serve as "YES", will advance processing to the following step SJ5, will move the display-position coordinate (X, Y) of a member's object OBJA code chosen by modification to a viewing-area A side, and will display as a new candidate's portrait image. Subsequently, at step SJ6, in response to a new candidate's portrait image being displayed, the display-position coordinate (X, Y) of a member's object OBJA code which separated by modification is moved to a viewing-area B side, and suppose that it is non-display.
[0091] The band member modification screen which illustrates actuation of the above-mentioned steps SJ5 and SJ6 to drawing 19 is mentioned as an example, and is explained. Suppose that three persons' portrait image which serves as a selection candidate is first displayed on the screen lower half side. And in order to display the portrait image which serves as a new candidate from this condition, actuation of changing a selection member is performed. The right key of the cross-joint key 109 or left key arranged in an equipment panel side is specifically operated, and a candidate is replaced. That is, when a right key is operated, a right end portrait image moves to a viewing-area B side, it becomes non-display, and the portrait image of a center and a left end shifts to right-hand side one by one according to this. Thereby, an opening is made at a left end, and a new candidate's portrait image is moved and displayed on this

location from a viewing-area B side. In addition, when a left key is operated, contrary to ****, the portrait image of a center and a right end acts as a left shift, respectively, and a new candidate is displayed on a right end location. Therefore, if it continues operating either of the right-and-left keys, it will left-shift-go round or right shift go round, and renewal of sequential of a candidate's portrait image will be carried out.

[0092] If modification of a selection member is made as mentioned above, it will judge whether actuation which CPU1 progresses to step SJ7 shown in drawing 34 , and makes a member decision was performed. Here, if the depression of the enter key 111 is carried out that a predetermined member (portrait image) should be determined, a decision result will serve as "YES" and processing will be advanced to the following step SJ8. At step SJ8, the display-position coordinate (X, Y) of the object OBJA code of the member who made a current decision, and the display-position coordinate (X, Y) of the object OBJA code of the member for modification are changed mutually. Thereby, the portrait image of the member who made a current decision is fixed to a viewing-area A side.

[0093] Subsequently, if it progresses to step SJ9, it judges whether the escape key 110 which restores a display condition was operated, and when the key 110 concerned is not operated, a decision result will serve as "NO", and this routine will once be completed. On the other hand, when an escape key 110 is operated in order to restore a display condition, a decision result serves as "YES" and processing is advanced to the following step SJ10. At step SJ10, the display-position coordinate (X, Y) of each object OBJA code of the portrait image currently displayed as a selection candidate is changed so that it may move to the viewing area B which becomes the outside of a screen display rectangle. And at the following step SJ11, the display-position coordinate (X, Y) of each object OBJA code of the selections display area evacuated to the viewing area B which becomes the outside of a screen display rectangle is changed so that it may display on the origin by the side of a viewing area A in a location. That is, at the above-mentioned steps SJ10-SJ11, if escape key actuation of restoring a display condition is made, while returning the portrait image currently displayed as a selection candidate to the viewing area B which becomes the outside of a screen display rectangle, each object of selections display area which evacuated to the viewing-area B side is moved to the original viewing-area A side, and it is set as a display condition. This changes from a band member modification screen to a band screen.

[0094] (g) If the object migration manipulation routine which an object color modification manipulation routine carries out **** of operation, and changes the display position of a portrait image according to modification of a band member is performed, CPU1 will perform the object color modification manipulation routine shown in drawing 35 through step SH2 (refer to drawing 32) mentioned above, and will advance processing to a step SK 1. It asks for the object which should cooperate to the processing of an object migration manipulation routine mentioned above, and should change a foreground color at a step SK 1. And according to the foreground color of the object set as the object of the color modification, the colour block number in object code is rewritten after a step SK 2. Hereafter, processing is explained according to the contents of color modification.

[0095] ** When changing into the Normal color, after completing band member modification actuation, in case it returns to a band screen, a color change of the object OBJA of a processing object is made at the Normal color showing a steady state. In this case, the decision result of a step SK 2 serves as "YES", it progresses to a step SK 3, and a colour block number with the Normal color of the target object OBJA is computed. And in the following step SK 4, the colour block number in the target object OBJA code is rewritten for the colour block number of the Normal color.

[0096] ** When changing into a selection color, in case a desired member is chosen at the time of band member modification, a color change of the object OBJA of a processing object is made at the selection color showing a selection condition. In this case, the decision result of a step SK 5 serves as "YES", it progresses to a step SK 6, and a colour block number with the selection color of the target object OBJA is computed. And in the following step SK 7, the colour block number in the target object OBJA code is rewritten for the colour block number of a selection color.

[0097] ** When changing into an active color, in case a desired member is determined at the time of band member modification, a color change of the object OBJA of a processing object is made at the

active color showing a decision condition. In this case, the decision result of the step SK 8 shown in drawing 36 serves as "YES", it progresses to a step SK 9, and a colour block number with the active color of the target object OBJA is computed. And in the following step SK 7, the colour block number in the target object OBJA code is rewritten for the colour block number of an active color.
[0098] Thus, in an object color modification manipulation routine, it asks for the object which should cooperate to the processing of an object migration manipulation routine mentioned above, and should change a foreground color, and the colour block number in the corresponding OBJA code is rewritten according to the transition state of the object for which it asked to what shows either the "Normal color", a "selection color" and an "active color." A part for the frame part of a portrait image and the PERT name part arranged at the lower part of a portrait image are changing the foreground color as an object which follows, for example, makes a color change according to a transition state. Even if it is in equipment without a mouse by this, the so-called GUI environment can be realized and it becomes possible to improve operability.
[0099] (h) As it is beyond actuation of a genre selection screen manipulation routine, if a band screen manipulation routine is completed, CPU1 will perform the genre selection screen manipulation routine shown in drawing 37 through the step SC 4 (refer to drawing 27) mentioned above, and will advance processing to a step SL 1. At a step SL 1, current Mohd judges whether it is under the genre selection mode which chooses a performance genre, when it is not that Mohd, a decision result here serves as "NO" and this routine is completed. On the other hand, if it is under genre selection mode, a decision result will serve as "YES" and processing will be advanced to the following step SL 2. At a step SL 2, the generated event judges whether it is the event which requires image modification.
[0100] The event which requires image modification in a genre selection screen points out the actuation which chooses the character string for example, in the window WIN in a genre selection screen (refer to drawing 20) according to the vertical key stroke of the cross-joint key 109, and actuation of newly carrying out an alphabetic character input into this selected character string part, and changing a character string. And when above-mentioned selection actuation or character string modification actuation is made, the decision result of a step SL 2 serves as "YES", and processing is advanced to a step SL 3. In addition, a decision result here becomes being the actuation which does not need image modification with "NO", and this routine is completed.
[0101] When the event which requires image modification is generated and processing is advanced to a step SL 3, it judges whether CPU1 is that in which the actuation makes a character string change. Here, a decision result serves as "NO" at the case which chooses the character string in Window WIN according to the vertical key stroke of the cross-joint key 109, and processing is advanced to a step SL 4. At a step SL 4, the colour block modification processing (OBJB) routine (refer to drawing 31) which changes the foreground color of the alphabetic character (object OBJB) directly above the cursor location which moves in the inside of Window WIN according to actuation of the cross-joint key 109 is performed.
[0102] That is, as mentioned above, by the colour block modification processing (OBJB) routine, cursor is located in the "alphabetic character" part in the Normal condition (black), the colour block number of the object code which corresponds so that the "alphabetic character" may be displayed in an active color (pink) when display modification assignment is carried out is rewritten, and when it changes display modification assignment into the Normal condition from an active state, the colour block number of the Normal color is assigned contrary to this.
[0103] On the other hand, when inputting a new alphabetic character into the character string which chose the character string in Window WIN according to the vertical key stroke of the cross-joint key 109 and changing a character string, the decision result of the above-mentioned step SL 3 serves as "YES", and CPU1 advances processing to a step SL 5. At a step SL 5, the character string modification manipulation routine (refer to drawing 30) mentioned above is performed. It is based on the cursor location scrolled according to actuation of the cross-joint key 109 when making a character string change in a genre selection screen. From the storage area of ROM2 to namely, 18 character strings The OBJB code for eight lines is read according to the value of the pointer register i. When the read OBJB

code is "a tooth space (null)", a transparence character (transparence color specification) is set up, when that is not right, it is set as the object B character (alphabetic character image) specified by the OBJB code, and this routine is completed.

[0104] (i) Actuation of a total control screen manipulation routine, next actuation of a total control screen manipulation routine are explained with reference to drawing 38 - drawing 40 . If an above-mentioned genre selection screen manipulation routine is completed, CPU1 will perform the total control screen manipulation routine shown in drawing 38 through a step SC 5 (refer to drawing 27), and will advance processing to a step SM 1. At a step SM 1, it judges whether it is under the total control mode to which current Mohd sets the mode of the whole equipment of operation, when it is not that Mohd, a decision result serves as "NO", and this routine is completed.

[0105] On the other hand, if it is under a total control mode, a decision result will serve as "YES" and processing will be advanced to the following step SM 2. At a step SM 2, the generated event judges whether it is the event which changes the image in a total control screen. Here, the event which changes an image points out the actuation which chooses the "carbon button" displayed on a total control screen (refer to drawing 23) according to actuation of the cross-joint key 109, and depression actuation of the enter key 111 made in case the mode of operation matched with this selected "carbon button" is set up. And if the cross-joint key 109 or an enter key 111 is operated, the decision result of a step SM 2 will serve as "YES", and processing will be advanced to a step SM 3. In addition, a decision result here becomes being the actuation which does not need image modification with "NO", and this routine is completed.

[0106] Now, the cross-joint key 109 or an enter key 111 is operated, when the event which requires image modification occurs, CPU1 advances processing to a step SM 3, and it judges whether the actuation is what changes the display condition of a "carbon button." Here, when it is the event from which the display condition of a "carbon button" does not change, modification processing of others, such as initialization which a decision result serves as "NO", and advances processing to the following step SM 4, for example, initializes a setup of the whole equipment, is performed, and this routine is completed. On the other hand, when the event which changes the display condition of a "carbon button" occurs The decision result of the above-mentioned step SM 3 serves as "YES", and CPU1 advances processing to a step SM 5. The below-mentioned colour block modification manipulation routine for carbon buttons is performed. The display condition of an applicable carbon button The foreground color of the object which forms a "carbon button" is changed that it should carry out to either "the Normal condition which is visible to a concave" and the "selection condition" which were illustrated to drawing 24 , and "the active state which is visible to convex."

[0107] (j) If the event which changes the display condition of a "carbon button" occurs as the colour block modification manipulation routine for carbon buttons carried out **** of operation, CPU1 will perform the colour block modification manipulation routine for carbon buttons shown in drawing 39 through a step SM 5. In addition, in order to attain simplification of explanation, colour block modification about the "BeepOn carbon button" is mentioned as an example among various carbon buttons, and it explains, and goes by this routine. If this routine is performed, CPU1 will advance processing to step SN1, and it will judge whether a processing object is the "BeepOn carbon button." If it is except the "BeepOn carbon button", a decision result will serve as "NO" and will progress to step SN2. And at step SN2, colour block modification processing about the carbon button of others other than the "BeepOn carbon button" is performed. On the other hand, when a processing object is the "BeepOn carbon button", the decision result of step SN1 serves as "YES", the below-mentioned BeepOn carbon button colour block modification manipulation routine is performed, the foreground color of a "carbon button" formation object is changed according to the actuation gestalt made about the carbon button concerned, and a carbon button display mode is changed.

[0108] (k) If the book routine of a BeepOn carbon button colour block modification manipulation routine of operation is performed, CPU1 will advance processing to the step SP 1 shown in drawing 40 , and will set up the target foreground color according to the actuation gestalt made about the BeepOn carbon button. Hereafter, it explains according to an actuation gestalt.

** Carbon button selection is made in operating the cross-joint key 109 at the time of carbon button selection, and locating cursor in either the on-carbon button or the off-carbon button, and in this case, the target foreground color turns into a "selection color" showing a selection condition, and advances processing to a step SP 2. At a step SP 2, the colour block number corresponding to this "selection color" is computed, and it stores in Register destCblock. Since a "selection color" is colour block #8 in color look-up table CLT4OBJA shown in drawing 16 , "8" is set to Register destCblock.
[0109] ** Carbon button decision is made by carrying out ON actuation of the enter key 111 in the above-mentioned carbon button selection condition at the time of carbon button decision (setup), and in this case, the target foreground color turns into an "active color" showing a decision (setup) condition, at the step SP 3 which advances processing to a step SP 3, it computes the colour block number corresponding to this "active color", and stores it in Register destCblock. Since an "active color" is colour block #7 in color look-up table CLT4OBJA shown in drawing 16 , "7" is set to Register destCblock.
[0110] ** When ON actuation of the enter key 111 is carried out at the time of carbon button decision at the time of setting cancellation, it becomes the "Normal color" which cancels a setup of the carbon button with which the actuation was made, and a pair of carbon button in exclusive relationship, and expresses a non-established state, and advance processing to a step SP 4. At a step SP 4, the colour block number corresponding to this the "Normal color" is computed, and it stores in Register destCblock. Since the "Normal color" is colour block #1 in color look-up table CLT4OBJA shown in drawing 16 , "1" is set to Register destCblock.
[0111] In this way, if the colour block number according to a display mode is set to Register destCblock, CPU1 will advance processing to a step SP 5. At a step SP 5, the object which should make a color change among object OBJA-A-OBJA-C (drawing 24 (**) reference) which constitutes the carbon button for modification (the on-carbon button or the off-carbon button) is extracted. Subsequently, at a step SP 6, the colour block value assigned to 4 bits of high orders of the OBJA code corresponding to the extracted object is rewritten for the colour block number in which it is stored by Register destCblock. Consequently, when made the Normal color, a display mode is changed so that it may be visible to convex, when a carbon button is made into an active color at a concave, and further, at the time of a selection color, a display mode is changed so that both in-between condition may be expressed.

[0112] (l) If the total control screen manipulation routine of a band save screen manipulation routine above-mentioned [of operation] is completed, CPU1 will perform the band save screen manipulation routine shown in drawing 41 through a step SC 6 (refer to drawing 27), and will advance processing to step SQ1. At step SQ1, current Mohd judges whether it is under the band save mode which carries out band registration, when it is not that Mohd, a decision result serves as "NO", and this routine is completed.
[0113] On the other hand, if it is under band save mode, a decision result will serve as "YES" and processing will be advanced to the following step SQ2. At step SQ2, the generated event judges whether it is the event which changes the image in a band save screen. In addition, the event which changes an image points out the actuation which chooses the character string for example, in the window WIN displayed on a band save screen (refer to drawing 22) according to actuation of the cross-joint key 109, and the character string modification actuation which rewrites the selected character string. And when these actuation is made, the decision result of step SQ2 is set to "YES", processing is advanced to step SQ3 and the other actuation is performed, a decision result serves as "NO" noting that it is the actuation which does not need image modification, and this routine is completed.
[0114] Now, the cross-joint key 109 or an enter key 111 is operated, and suppose that the event which requires image modification occurred. If it does so, it will judge whether CPU1 is that in which advances processing to step SQ3 and the actuation makes "character string selection." Here, when "character string selection" is performed by the vertical key stroke of the cross-joint key 109, a decision result serves as "YES" and processing is advanced to step SQ4. At step SQ4, the foreground color of the object OBJA which performs the below-mentioned colour block modification processing (OBJA)

routine, and is put on the same location as the selected character string (object OBJB) is changed into a "selection color", a selection condition is shown, and this routine is once completed.
[0115] On the other hand, when actuation other than "character string selection" is performed, the decision result of the above-mentioned step SQ3 serves as "NO", and processing is advanced to step SQ5. At step SQ5, when it judges whether actuation which rewrites the selected character string was performed and such actuation is not made, processing is advanced to step SQ6 by setting a decision result to "NO", and processing corresponding to the actuation is performed. On the other hand, when actuation which rewrites the selected character string is performed, the decision result of step SQ5 serves as "YES", and processing is advanced to step SQ7.
[0116] If it progresses to step SQ7, CPU1 will perform the character string modification manipulation routine (refer to drawing 30) mentioned above. It is based on the cursor location scrolled according to actuation of the cross-joint key 109 in this character string modification manipulation routine. From the storage area of ROM2 to 18 character strings The OBJB code for eight lines is read according to the value of the pointer register i. When the read OBJB code is "a tooth space (null)", a transparence character (transparence color specification) is set up, and when that is not right, it is set as the object B character (alphabetic character image) specified by the OBJB code. And perform the colour block modification processing (OBJB) routine (refer to drawing 31) mentioned above, when an object object is the Normal color, an active color is made to make a color change, and when contrary to this, the Normal color is made to make a color change of CPU1, if it progresses to step SQ8.
[0117] (m) If the colour block modification processing (OBJA) routine shown in drawing 42 through the step SQ4 of a colour block modification processing (OBJA) routine above-mentioned [of operation] is performed, CPU1 will advance processing to a step SR 1. First, it asks for the object object which agrees with the cursor location moved according to actuation of the cross-joint key 109 at a step SR 1. And according to the foreground color of the object set as the object of the color modification, the colour block number in object code is rewritten after a step SR 2. Hereafter, processing is explained according to the contents of color modification.
[0118] ** When changing into the Normal color, a color change of the object OBJA from which it separated for selection by actuation of the cross-joint key 109 is made at the Normal color showing a steady state. In this case, the decision result of a step SR 2 serves as "YES", it progresses to a step SR 3, and a colour block number with the Normal color of the target object OBJA is computed. And in the following step SR 4, the colour block number in the target object OBJA code is rewritten for the colour block number of the Normal color.
[0119] ** When changing into a selection color, a color change of the object OBJA made applicable to selection by actuation of the cross-joint key 109 is made at the selection color showing a selection condition. In this case, the decision result of a step SR 5 serves as "YES", it progresses to a step SR 6, and a colour block number with the selection color of the target object OBJA is computed. And in the following step SR 7, the colour block number in the target object OBJA code is rewritten for the colour block number of a selection color.
[0120] Thus, while the foreground color of the object OBJA made applicable to selection by actuation of the cross-joint key 109 is changed into a "selection color" and a selection condition is shown, he is trying to return the object OBJA from which it separated for selection to the "Normal color" by the colour block modification processing (OBJA) routine.
[0121] (n) Actuation, now CPU1 of V blank interrupt manipulation routine are the process in which each routine explained until now is performed, and V blank interrupt manipulation routine which carries out the DMA transfer of object code required for a screen display, image data, and the color look-up table to the VDP5 and VRAM6 side synchronizing with the vertical-retrace-line period by the side of Display DP is performed. Whenever it enters at a vertical-retrace-line period, interruption activation is carried out, and this routine carries out the DMA transfer of the object code corresponding to screen mode, image data, and color look-up table at that time to the VDP5 and VRAM6 side, and, below, gives explanation of operation according to each screen mode.
[0122] ** initial-screen Mohd -- if interruption activation of this routine is first carried out whenever it

enters at a vertical-retrace-line period, CPU1 will advance processing to the step SS 1 shown in drawing 43 . At a step SS 1, it judges whether current screen mode is initial-screen Mohd. And a decision result becomes being initial-screen Mohd with "YES", and processing is advanced to the following step SS 2. At a step SS 2, it judges whether the initial-screen transfer flag stored in Register IGF is "1." Since "1" is set when it changes to an initial screen in the screen change manipulation routine (refer to drawing 28) mentioned above, a decision result serves as "YES" and an initial-screen transfer flag advances processing to the following step SS 3, when changing to the initial screen.

[0123] At a step SS 3, the DMA transfer of background image data IM1BG and object A image data IM1OBJA (refer to drawing 7) which form an initial screen is carried out from ROM2 to VRAM6. Subsequently, if it progresses to a step SS 4, the DMA transfer of the above-mentioned image data IM1BG, color look-up table CLT1BG matched with IM1OBJA, and the CLT1OBJA will be carried out to the table area CLTA for object A of the color look-up table section 65 (refer to drawing 17) from ROM2. And it progresses to a step SS 5, and zero reset of the initial-screen transfer flag stored in Register IGF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 6.

[0124] At a step SS 6, OBJA code OBJ1OBJA for initial screens stored in RAM3 by the screen change manipulation routine mentioned above is read, and it transmits to the object memory 52 of the VDP5 interior. Consequently, VDP5 generates the initial screen which should be displayed with degree frame. In addition, when zero reset of the initial-screen transfer flag which a transfer is already completed and is stored in Register IGF is carried out when the decision result of the above-mentioned step SS 2 is "NO" namely, the above-mentioned step SS 6 is performed and OBJA code OBJ1OBJA for initial screens is transmitted to the object memory (OM) 52.

[0125] ** band screen mode -- in this case, the decision result of a step SS 7 serves as "YES", and advance processing to a step SS 8. At a step SS 8, it judges whether the band screen transfer flag stored in Register BGF is "1." Since "1" is set when it changes to a band screen in a screen change manipulation routine (refer to drawing 28), when changing to the band screen, a decision result serves as "YES", and a band screen transfer flag advances processing to the following step SS 9. At a step SS 9, the DMA transfer of background image data IM2BG which forms a band screen, and the object A image data IM2OBJA (refer to drawing 7) is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 10, the DMA transfer of the object B image data IM2OBJB which forms a band screen is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5 interior.

[0126] Subsequently, if it progresses to a step SS 11, CPU1 will carry out the DMA transfer of above-mentioned image data IM2BG, color look-up table CLT2BG respectively matched with IM2OBJA, and the CLT2OBJA to the table area CLTA for object A of the color look-up table section 65 (refer to drawing 17) from ROM2. Then, at a step SS 12, the DMA transfer of the color look-up table CLT2OBJB matched with image data IM2OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 from ROM2. And it progresses to a step SS 13 next, and zero reset of the band screen transfer flag stored in Register BGF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 14.

[0127] If it progresses to a step SS 14, CPU1 reads OBJA code OBJ2OBJA for band screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 15, will read OBJB code OBJ2OBJB for band screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the band screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the band screen transfer flag is carried out when the decision result of the above-mentioned step SS 8 is "NO" namely, the above-mentioned steps SS14 and SS15 are performed, and OBJA code OBJ2OBJA for band screens and OBJB code OBJ2OBJB for band screens are transmitted to the object memory (OM) 52, respectively.

[0128] ** genre selection screen mode -- the decision result of the step SS 16 shown in drawing 44 serves as "YES" in this case, and advance processing to a step SS 17. At a step SS 17, it judges whether the genre selection screen transfer flag stored in Register JSF is "1." Since this genre selection screen

transfer flag is set to "1" when it is changed to a genre selection screen, a decision result here serves as "YES" and advances processing to the following step SS 18. At a step SS 18, the DMA transfer of background image data IM3BG and object A image data IM3OBJA (refer to drawing 7) which form a genre selection screen is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 19, and it forms a genre selection screen, the DMA transfer of the object B image data IM3OBJB is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5 interior.

[0129] Subsequently, if it progresses to a step SS 20, CPU1 will carry out the DMA transfer of the above-mentioned image data IM3BG, color look-up table CLT3BG respectively matched with IM3OBJA, and the CLT3OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. Then, at a step SS 21, the DMA transfer of the color look-up table CLT3OBJB matched with image data IM3OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 22 next, and zero reset of the band screen transfer flag stored in Register JSF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 23.

[0130] If it progresses to a step SS 23, CPU1 reads OBJA code OBJ3OBJA for genre selection screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 24, will read OBJB code OBJ3OBJB for genre selection screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the genre selection screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the genre selection screen transfer flag is carried out when the decision result of the above-mentioned step SS 17 is "NO" namely, the above-mentioned steps SS23 and SS24 are performed, and OBJA code OBJ3OBJA for genre selection screens and OBJB code OBJ3OBJB for genre selection screens are transmitted to the object memory (OM) 52.

[0131] ** total control screen mode -- in this case, the decision result of a step SS 25 serves as "YES", and advance processing to a step SS 26. At a step SS 26, it judges whether the total control screen transfer flag stored in Register TCF is "1." Since this flag is set to "1" when it is changed to a total control screen, a decision result here serves as "YES" and advances processing to the following step SS 27. At a step SS 27, the DMA transfer of background image data IM4BG and object A image data IM4OBJA (refer to drawing 7) which form a total control screen is carried out from ROM2 to VRAM6, respectively.

[0132] Subsequently, if it progresses to a step SS 28, CPU1 will carry out the DMA transfer of the above-mentioned image data IM4BG, color look-up table CLT4BG respectively matched with IM4OBJA, and the CLT4OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 29 next, and zero reset of the total control screen transfer flag stored in Register TCF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 30. At a step SS 30, OBJA code OBJ4OBJA for total control screens stored in RAM3 by the screen change manipulation routine mentioned above is read, and it transmits to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the total control screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the total control screen transfer flag is carried out when the decision result of the above-mentioned step SS 25 is "NO" namely, the above-mentioned step SS 30 is performed and OBJA code OBJ4OBJA for total control screens is transmitted to the object memory (OM) 52.

[0133] ** band save screen mode -- in this case, the decision result of a step SS 31 serves as "YES", and advance processing to a step SS 32. At a step SS 32, it judges whether the band save screen transfer flag stored in Register BSF is "1." Since this transfer flag is set to "1" when it is changed to a band save screen, a decision result here serves as "YES" and advances processing to the following step SS 33. At a step SS 33, the DMA transfer of background image data IM5BG and object A image data IM5OBJA (refer to drawing 7) which form a band save screen is carried out from ROM2 to VRAM6, respectively. Then, at a step SS 34, and it forms a band save screen, the DMA transfer of the object B image data IM5OBJB is carried out from ROM2 to the character generator memory (CGM) 54 of the VDP5

interior.
[0134] Subsequently, if it progresses to a step SS 35, CPU1 will carry out the DMA transfer of the above-mentioned image data IM5BG, color look-up table CLT5BG respectively matched with IM5OBJA, and the CLT5OBJA to the table area CLTA for object A of the color look-up table section 65 interior from ROM2. Then, at a step SS 36, the DMA transfer of the color look-up table CLT5OBJB matched with image data IM5OBJB is carried out to the table area CLTB for object B of the color look-up table section 65 interior from ROM2. And it progresses to a step SS 37 next, and zero reset of the band save screen transfer flag stored in Register BSF is carried out, the completion of a transfer is expressed, and processing is advanced to a step SS 38.
[0135] If it progresses to a step SS 38, CPU1 reads OBJA code OBJ6OBJA for band save screens stored in RAM3 by the screen change manipulation routine mentioned above, transmits it to the object memory (OM) 52 of the VDP5 interior, in the continuing step SS 39, will read OBJB code OBJ6OBJB for band save screens from RAM3 similarly, and will transmit it to the object memory (OM) 52 of the VDP5 interior. Thereby, VDP5 generates the band save screen which should be displayed with degree frame. In addition, when a transfer is already completed and zero reset of the band save screen transfer flag is carried out when the decision result of the above-mentioned step SS 32 is "NO" namely, the above-mentioned steps SS38 and SS39 are performed, and OBJA code OBJ6OBJA for band save screens and OBJB code OBJ6OBJB for band save screens are transmitted to the object memory (OM) 52.
[0136] As explained above, while forming the alphabetic character displayed fixed on screens, such as a screen title, in this example by the object A image To the viewing area (display line) which displays the alphabetic character by which a display change is made on this object A image The object B image for several maximum alphabetic character minutes which can be displayed beforehand is arranged. When rewriting the alphabetic character, in case only the character name in the OBJB code corresponding to the display position (alphabetic character class) is changed and an alphabetic character is eliminated, transparence assignment of the colour block in the OBJB code is carried out, and it is changing into the non-display condition. Therefore, like the conventional image control unit, about both alphabetic character by which a display change is made on a screen, and alphabetic characters displayed fixed, such as a screen title, the class and display coordinate are managed separately, or the processing which moves the display position in corresponding object code out of a screen-display field becomes unnecessary in the case of character deletion. For this reason, if it becomes possible to aim at improvement in processing speed and this is put in another way, without using high-speed CPU, improvement in processing speed is attained, without inviting a cost rise.
[0137] If it is in the example mentioned above, it does not have pointing devices, such as a mouse. Moreover, ** The object A image which should change a foreground color according to actuation of the cross-joint key 109 or ENTA 111 is specified. Since the colour block number in the OBJA code corresponding to this was rewritten to what shows either the "Normal color", a "selection color" and an "active color" Improvement in operability, such as being able to realize the GUI environment which makes an image display mode adjustable corresponding to the contents of a key stroke, consequently preventing an operation mistake, is achieved. As this and a similar point, in the example mentioned above, the "carbon button" by which a screen display is carried out was constituted from object OBJA-A-OBJA-C (drawing 24 (**) reference), and in order to change the foreground color of these object OBJA-A-OBJA-C according to the contents of the actuation gestalt which makes an image change and to display the condition (the Normal condition, a selection condition, and active state) of relevance "a carbon button", the GUI environment excellent in visibility is attained.
[0138] Furthermore, the reversal flag X is formed in the OBJA code of the object A image displayed fixed on a screen in the example mentioned above. [whether the order of the write-in sequence at the time of storing object A image data in a line buffer according to the value of this flag X of the address is carried out, and] Or since it enabled it to specify whether it is made it and a reverse order, it is made to display as a usual image which wrote in one image data in order of the address, or it becomes possible to display the reverse image which wrote this in the reverse order and right and left of an image reversed. That is, the class of image which it can display since two images with a thing can be formed [which

shall inverse-video-display one image data on usual] now can use image data efficiently for a system with little increase and memory space.

[0139] In addition, in this example, although reference was made about the case where it applies to the musical-sound control device by which the image control device by this invention is connected to a display, needless to say, the summary of this invention is not limited to this, for example, information machines and equipment, such as a personal computer and a word processor, can be applied also to personal digital assistant equipments (a pager, PHS, etc.) or video game equipment from the first. If it is equipment which realizes the user interface using an image in short, the summary of this invention can be applied and it excels in visibility by doing so, and the remarkable effectiveness of making the processing speed of a display control improve without inviting the cost rise of a product is acquired, realizing the operating environment which prevents an operation mistake.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the external view showing the appearance of one example by this invention.
[Drawing 2] It is the block diagram showing this whole example configuration.
[Drawing 3] It is a memory map for explaining the object code class stored in ROM2 in this example.
[Drawing 4] It is a memory map for explaining the object code class stored in ROM2 in this example.
[Drawing 5] It is drawing showing the configuration of the OBJA code stored in ROM2.
[Drawing 6] It is drawing showing the configuration of the OBJB code stored in ROM2.
[Drawing 7] It is a memory map for explaining the image kind of data stored in ROM2 in this example.
[Drawing 8] It is drawing showing the configuration of the object B image data stored in ROM2.
[Drawing 9] It is drawing showing the configuration of the object B image data stored in ROM2.
[Drawing 10] It is drawing showing the configuration of the object A image data stored in ROM2.
[Drawing 11] It is drawing showing the configuration of the object A image data stored in ROM2.
[Drawing 12] It is a memory map for explaining the color look-up table class stored in ROM2 in this example.
[Drawing 13] It is drawing showing the configuration of color look-up table CLT2OBJA for object OBJA used for a band screen.
[Drawing 14] It is drawing showing the configuration of color look-up table CLT3OBJA for object OBJA used for a genre selection screen.
[Drawing 15] It is drawing showing the configuration of color look-up table CLT3OBJB for object OBJB used for a genre selection screen.
[Drawing 16] It is drawing showing the configuration of color look-up table CLT4OBJA for object OBJA used for a total control screen.
[Drawing 17] It is the block diagram showing the configuration of VDP5 in this example.
[Drawing 18] It is drawing showing the example of a configuration of a band screen.
[Drawing 19] It is drawing showing the example of a configuration of a band member modification screen.
[Drawing 20] It is drawing showing the example of a configuration of a genre selection screen.
[Drawing 21] It is drawing for explaining the object class which constitutes the window WIN displayed on a genre selection screen.
[Drawing 22] It is drawing showing the example of a configuration of a band save screen.
[Drawing 23] It is drawing showing the example of a configuration of a total control screen.
[Drawing 24] It is drawing showing the display gestalt and object class of object which constitute the carbon button displayed on a total control screen.
[Drawing 25] It is the flow chart which shows outline actuation of this example.
[Drawing 26] It is a flow chart for explaining the Maine flow in the above-mentioned outline actuation.
[Drawing 27] It is a flow chart for explaining the contents of the image processing in the Maine flow.
[Drawing 28] It is a flow chart for explaining actuation of the screen change manipulation routine in an image processing.

[Drawing 29] It is a flow chart for explaining actuation of the band screen manipulation routine in an image processing.
[Drawing 30] It is a flow chart for explaining actuation of the character string modification manipulation routine in an image processing.
[Drawing 31] It is a flow chart for explaining actuation of the colour block modification processing (OBJB) routine in an image processing.
[Drawing 32] It is a flow chart for explaining actuation of the band member modification manipulation routine in an image processing.
[Drawing 33] It is a flow chart for explaining actuation of the object migration manipulation routine in an image processing.
[Drawing 34] It is a flow chart for explaining actuation of the object migration manipulation routine in an image processing.
[Drawing 35] It is a flow chart for explaining actuation of the object color modification manipulation routine in an image processing.
[Drawing 36] It is a flow chart for explaining actuation of the object color modification manipulation routine in an image processing.
[Drawing 37] It is a flow chart for explaining actuation of the genre selection screen manipulation routine in an image processing.
[Drawing 38] It is a flow chart for explaining actuation of the total control screen manipulation routine in an image processing.
[Drawing 39] It is a flow chart for explaining actuation of the colour block modification manipulation routine for carbon buttons in an image processing.
[Drawing 40] It is a flow chart for explaining actuation of the BeepOn carbon button colour block modification manipulation routine in an image processing.
[Drawing 41] It is a flow chart for explaining actuation of the band save screen manipulation routine in an image processing.
[Drawing 42] It is a flow chart for explaining actuation of the colour block modification processing (OBJA) routine in an image processing.
[Drawing 43] It is a flow chart for explaining actuation of V blank interrupt manipulation routine in an image processing.
[Drawing 44] It is a flow chart for explaining actuation of V blank interrupt manipulation routine in an image processing.
[Description of Notations]
1 CPU (Image Arrangement Means, Modification Means, Display-Control Means)
2 ROM (Image Arrangement Means, Display-Control Means)
3 RAM (Display-Control Means)
4 Keyboard and Switch I/F Circuit
5 VDP
6 VRAM
7 Encoder
8 Sound Source
9 D/A Converter
101 Keyboard
103-111 Panel switch (modification means)

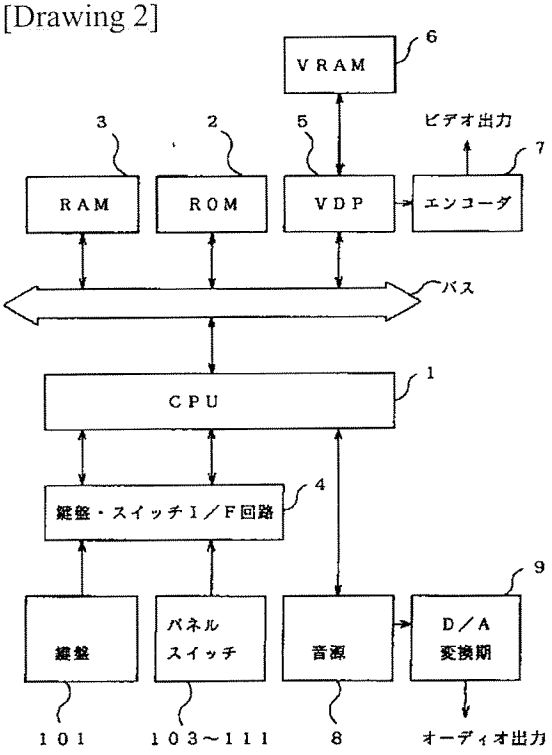
[Translation done.]

* NOTICES *

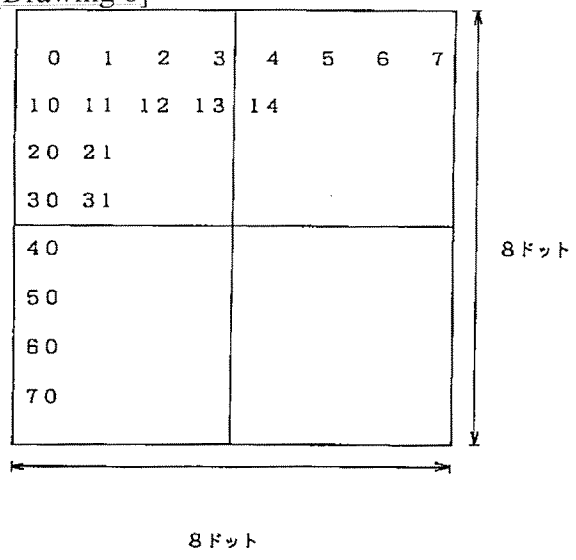
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

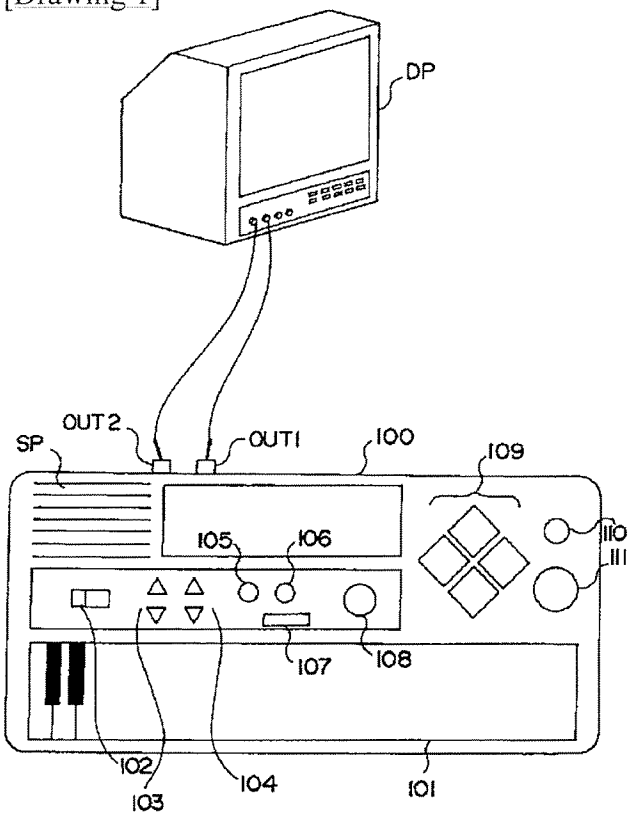
[Drawing 2]



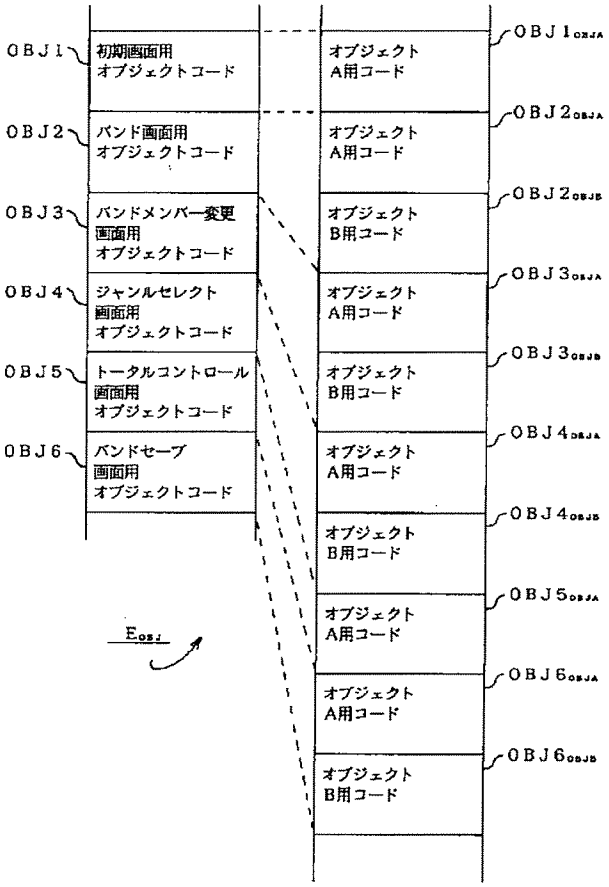
[Drawing 8]



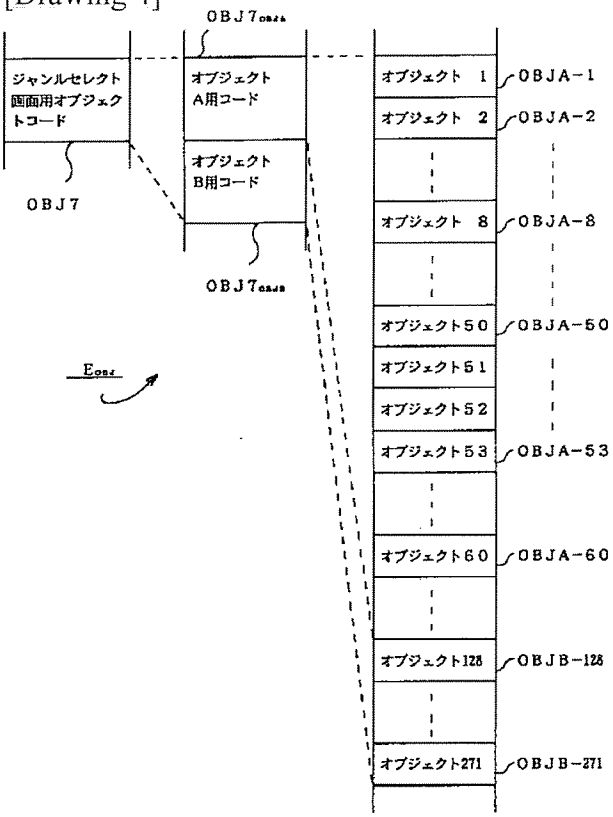
[Drawing 1]



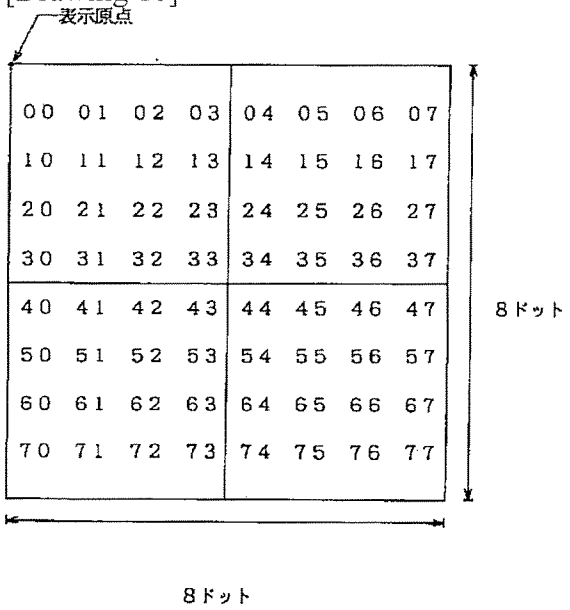
[Drawing 3]


OBJ1
初期画面用
オブジェクトコード
OBJ2
バンド画面用
オブジェクトコード
OBJ3
バンドメンバー変更
画面用
オブジェクトコード
OBJ4
ジャンルセレクト
画面用
オブジェクトコード
OBJ5
トータルコントロール
画面用
オブジェクトコード
OBJ6
バンドセーブ
画面用
オブジェクトコード
オブジェクト
A用コード
オブジェクト
B用コード
OBJ1OBJA
OBJ2OBJA
OBJ2OBJB
OBJ3OBJA
OBJ3OBJB
OBJ4OBJA
OBJ4OBJB
OBJ5OBJA
OBJ6OBJA
OBJ6OBJB

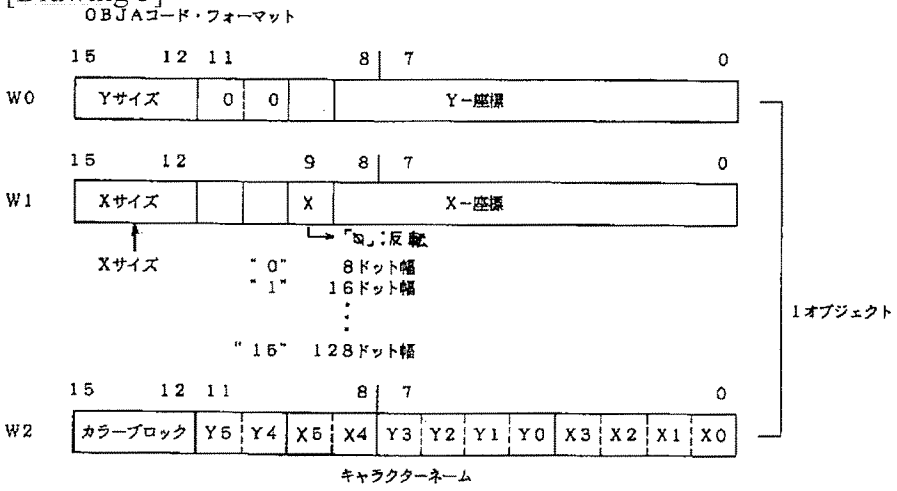
[Drawing 4]


OBJ7OBJA
ジャンルセレクト
画面用オブジェク
トコード
OBJ7
オブジェクト
A用コード
オブジェクト
B用コード
OBJ7OBJB
オブジェクト 1
オブジェクト 2
オブジェクト 8
オブジェクト50
オブジェクト51
オブジェクト52
オブジェクト53
オブジェクト60
オブジェクト128
オブジェクト271
OBJA-1
OBJA-2
OBJA-8
OBJA-50
OBJA-53
OBJA-60
OBJB-128
OBJB-271

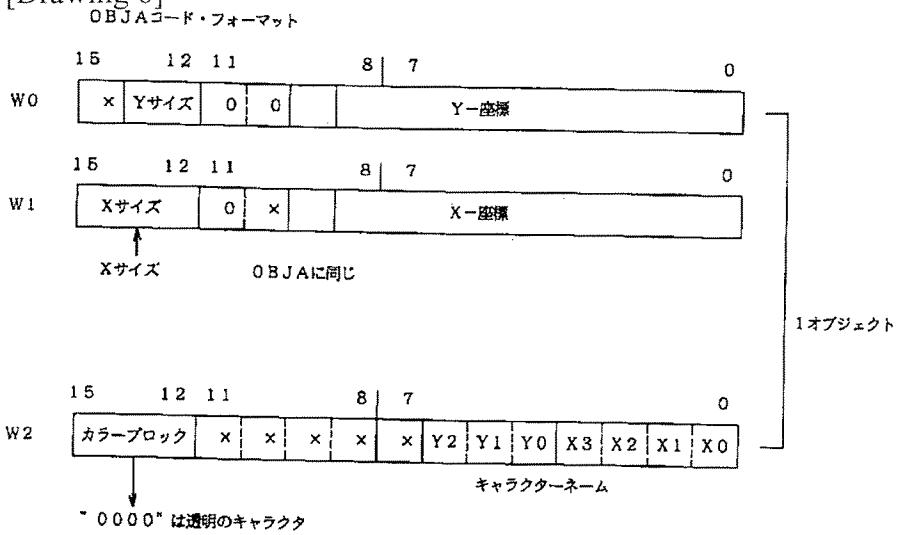
[Drawing 10]



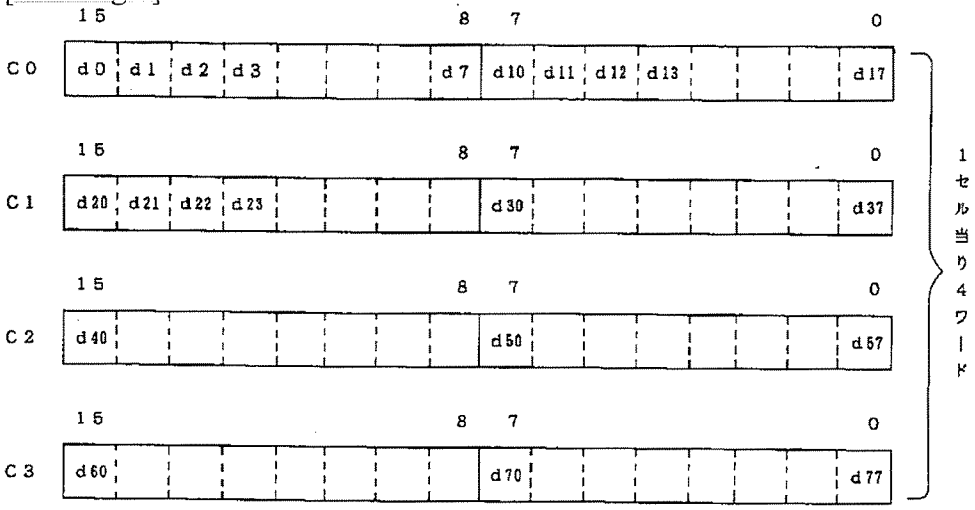
[Drawing 5]



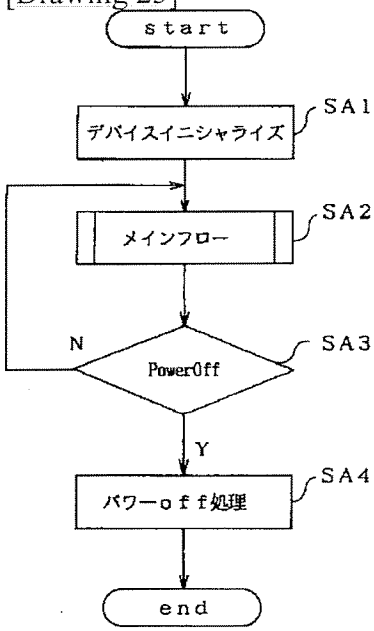
[Drawing 6]



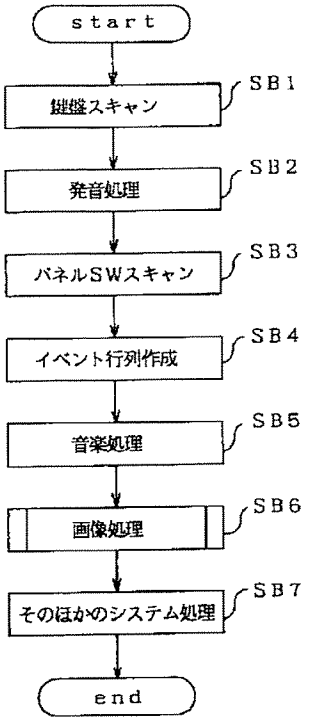
[Drawing 9]



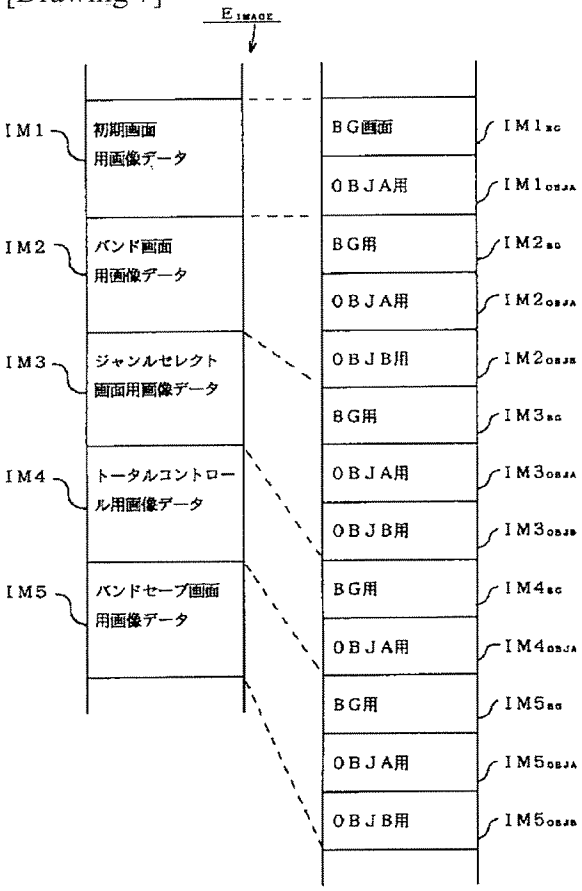
[Drawing 25]



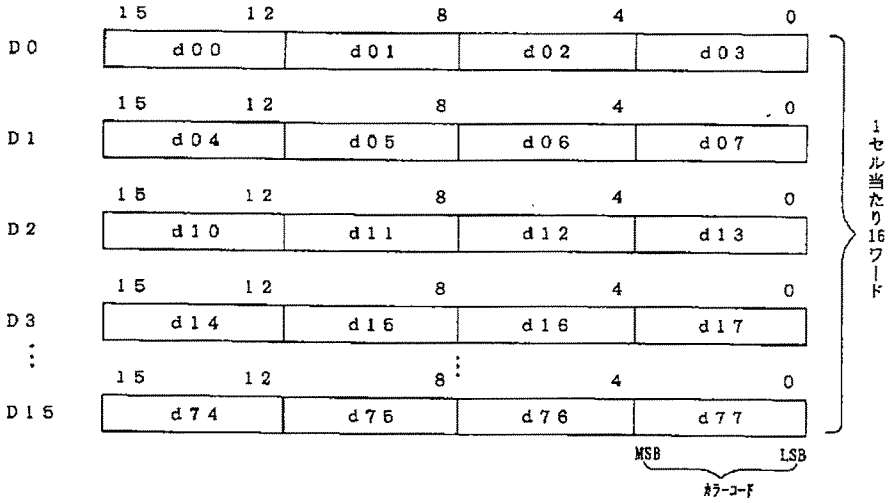
[Drawing 26]



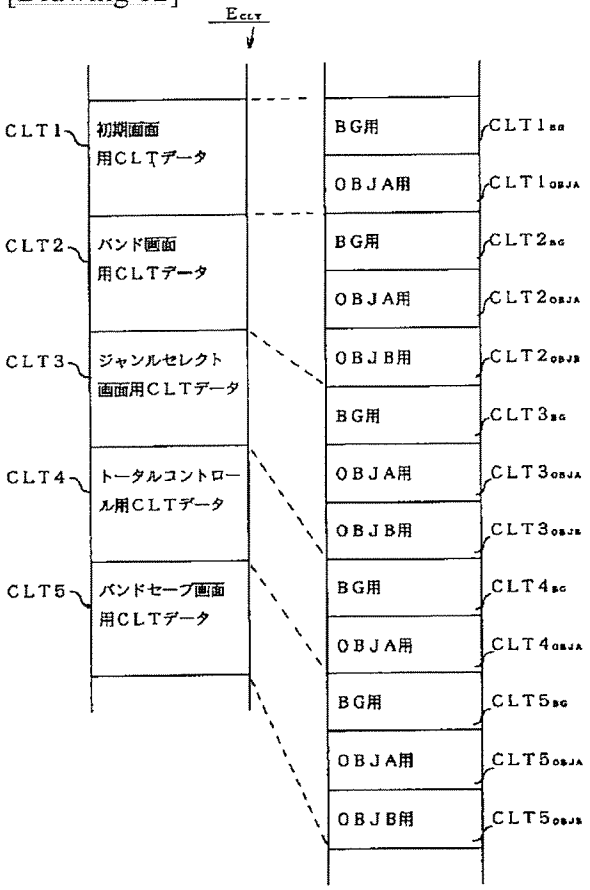
[Drawing 7]



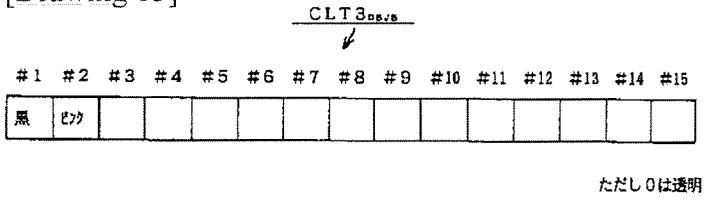
[Drawing 11]



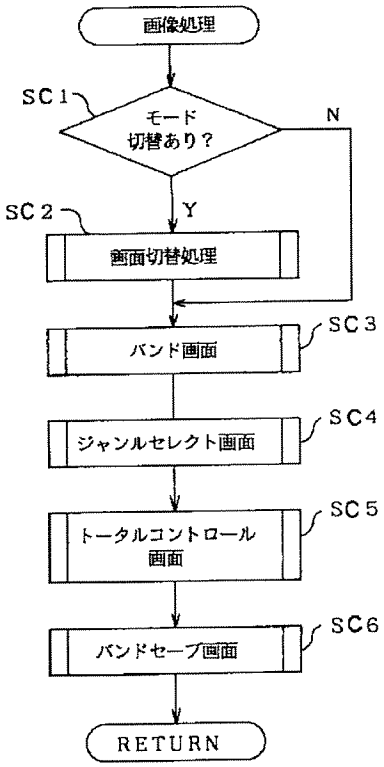
[Drawing 12]



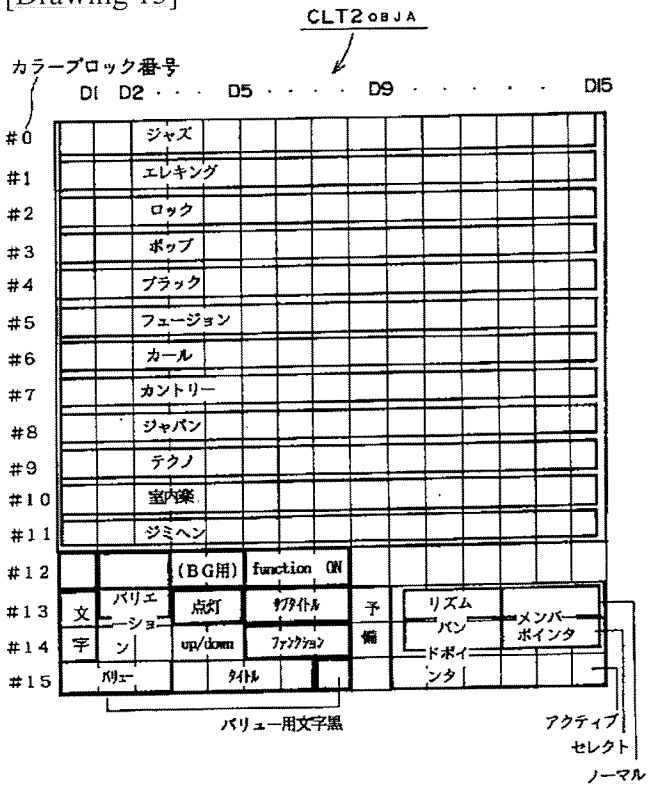
[Drawing 15]



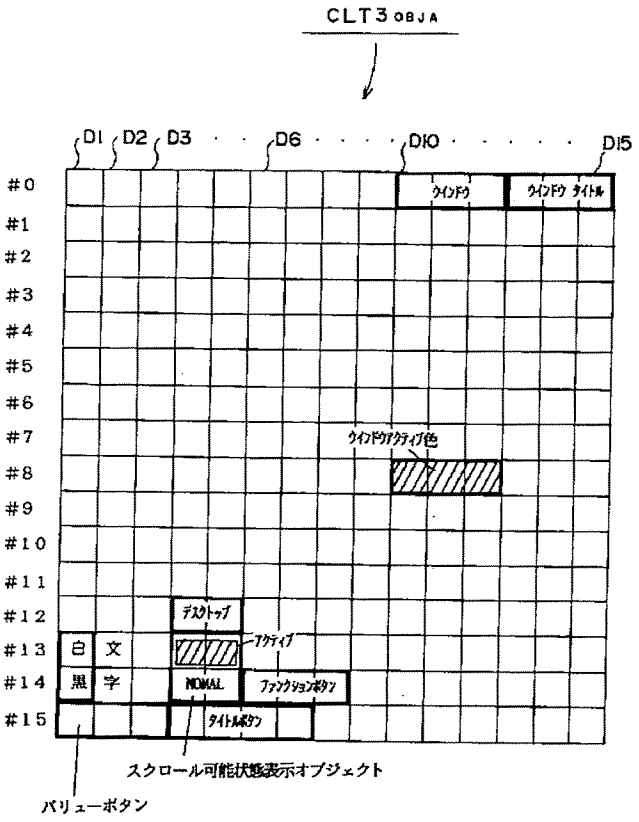
[Drawing 27]



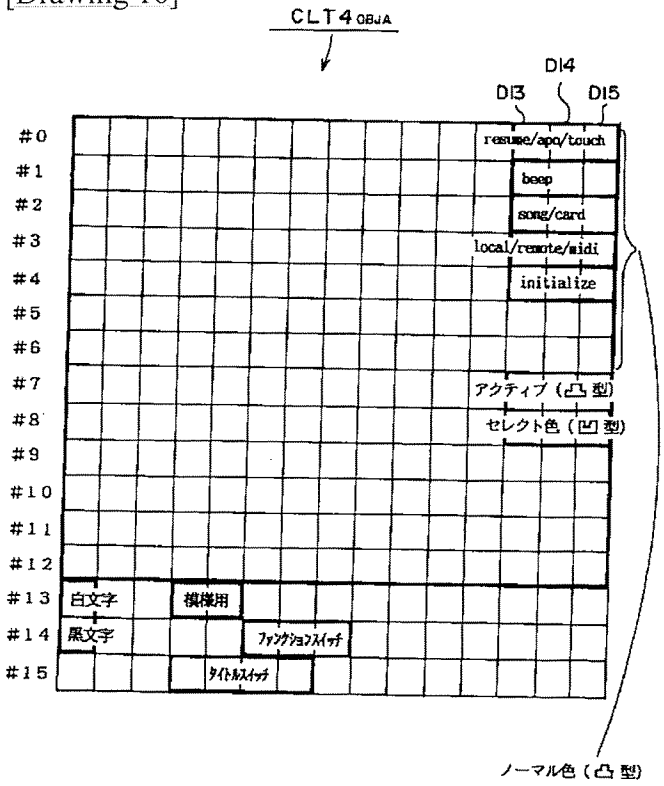
[Drawing 13]



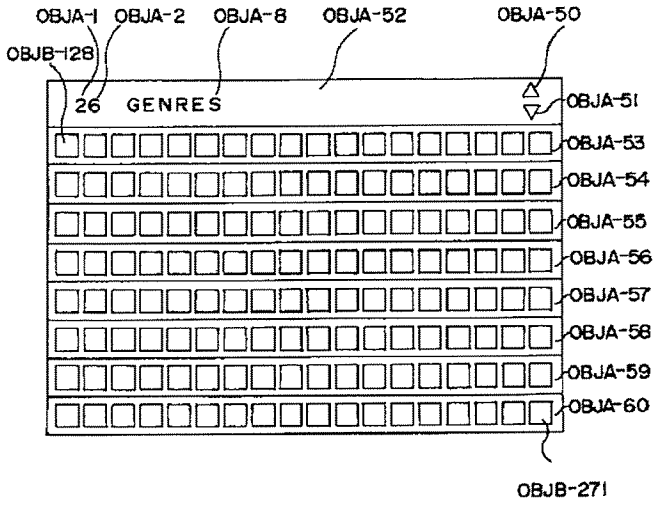
[Drawing 14]



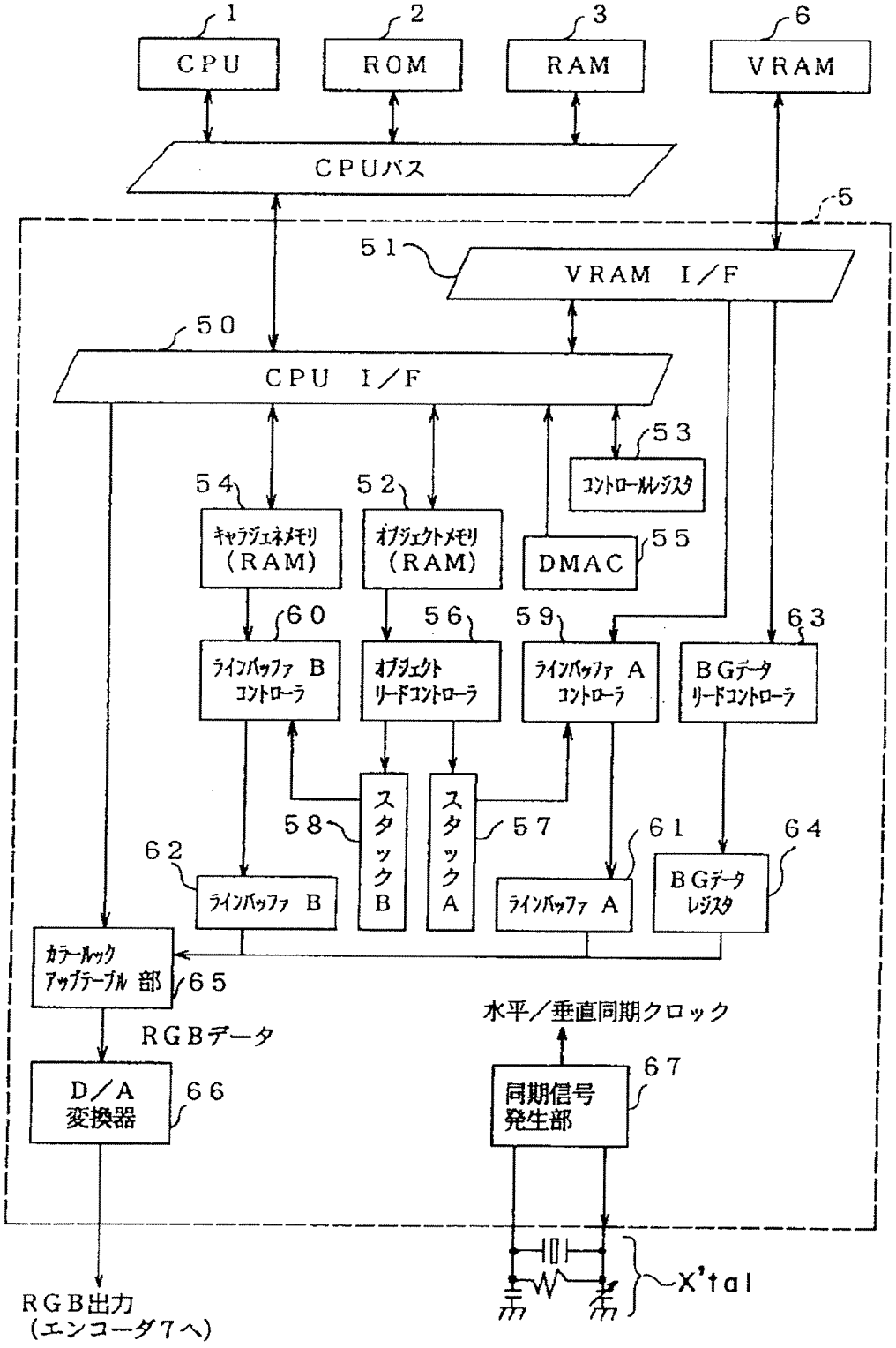
[Drawing 16]



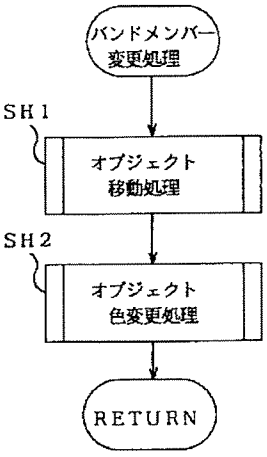
[Drawing 21]


OBJA-1
OBJA-2
OBJA-8
OBJA-52
OBJA-50
OBJB-128
26
GENRES
OBJA-51
OBJA-53
OBJA-54
OBJA-55
OBJA-56
OBJA-57
OBJA-58
OBJA-59
OBJA-60
OBJB-271

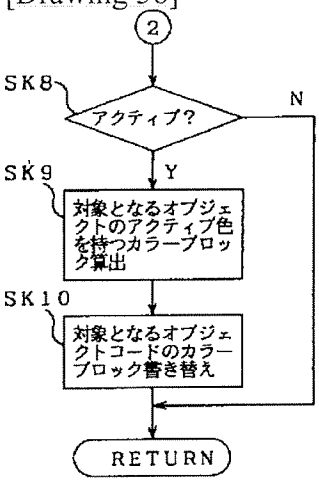
[Drawing 17]


1
CPU
2
ROM
3
RAM
6
VRAM
CPUバス
5
51
VRAM I／F
50
CPU I／F
53
コントロールレジスタ
54
キャラジェネメモリ (RAM)
52
オブジェクトメモリ (RAM)
DMAC
55
60
ラインバッファ B コントローラ
56
オブジェクト リードコントローラ
59
ラインバッファ A コントローラ
63
BGデータ リードコントローラ
58
スタックB
57
スタックA
62
ラインバッファ B
61
ラインバッファ A
64
BGデータ レジスタ
カラールックアップテーブル部
65
RGBデータ
D／A 変換器
66
水平／垂直同期クロック
67
同期信号 発生部
X'tal
RGB出力 (エンコーダ7へ)

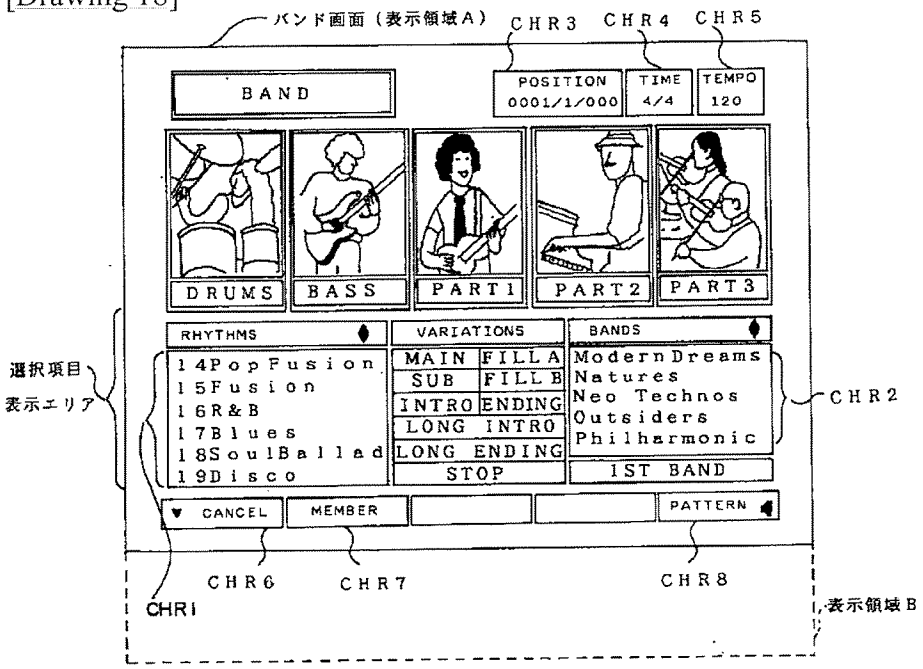
[Drawing 32]



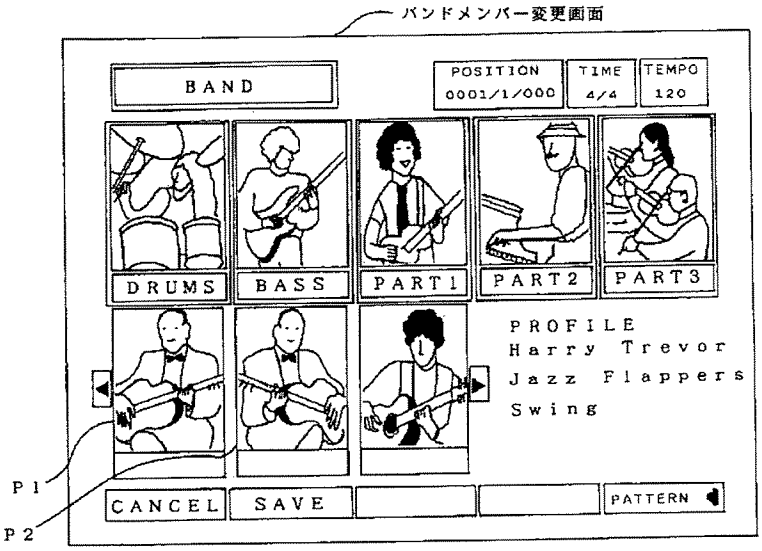
[Drawing 36]



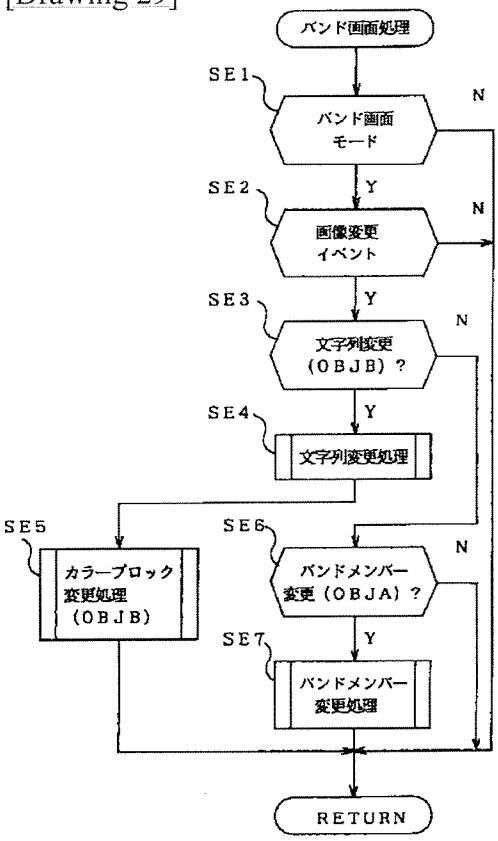
[Drawing 18]



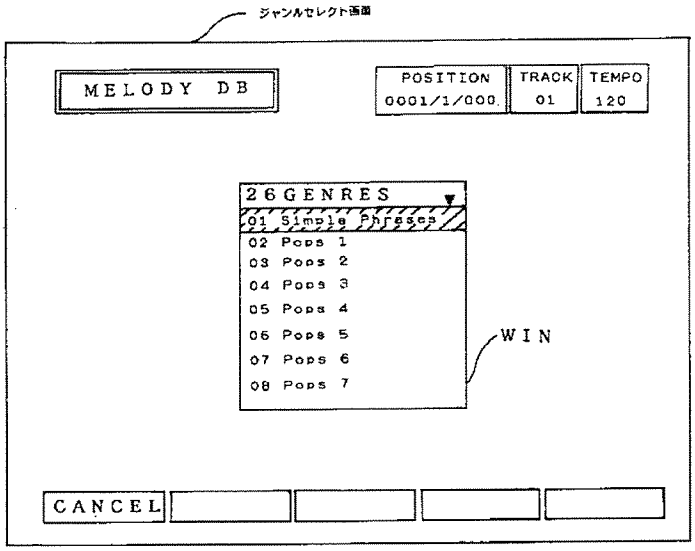
[Drawing 19]



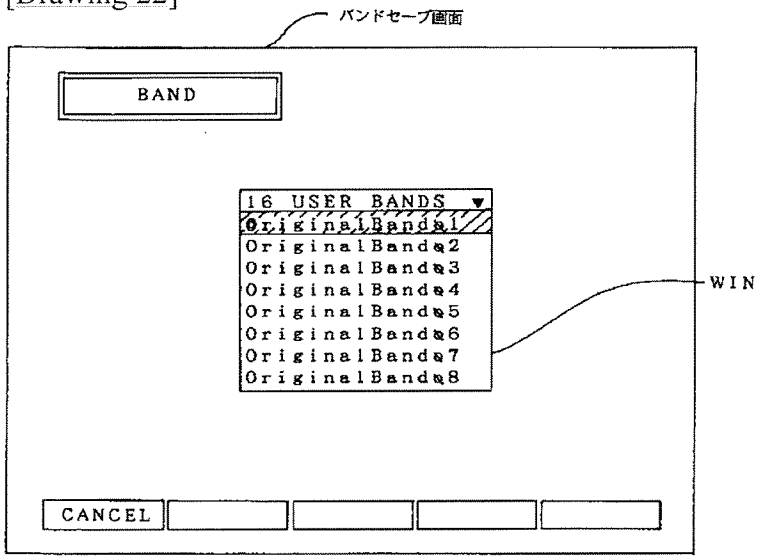
[Drawing 29]



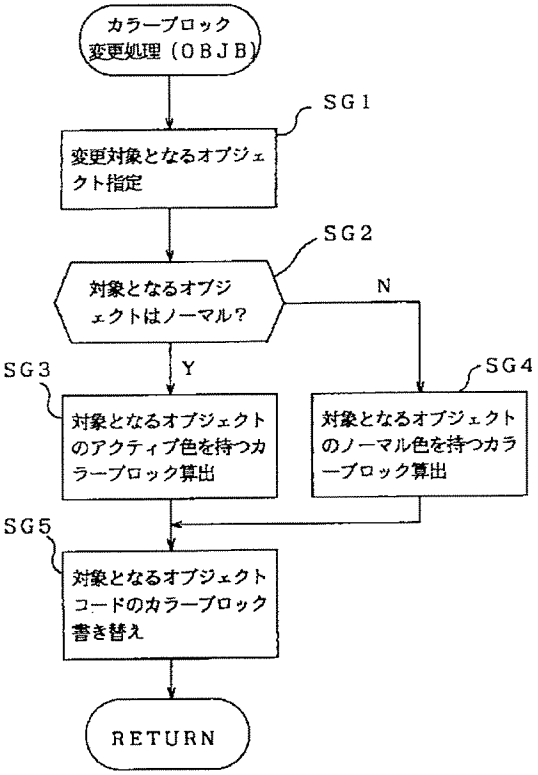
[Drawing 20]



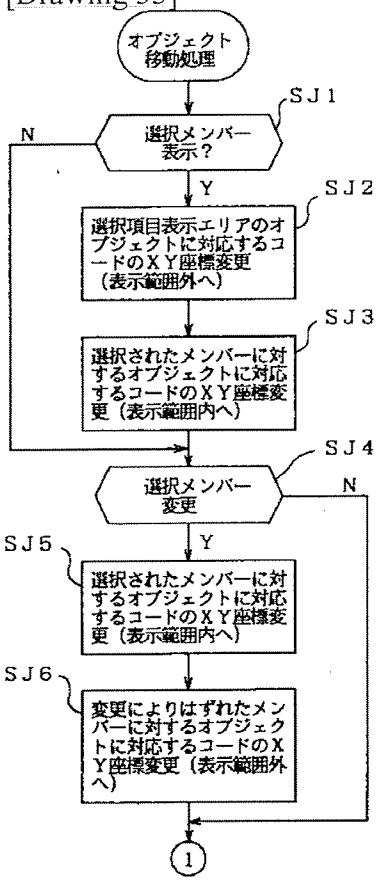
[Drawing 22]



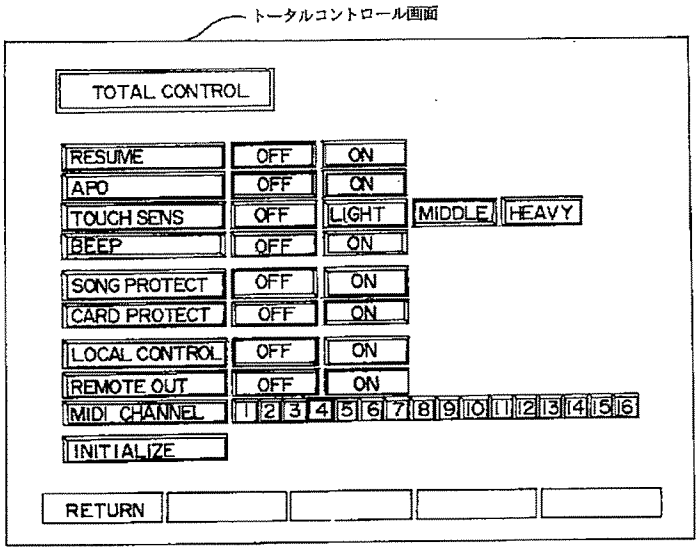
[Drawing 31]



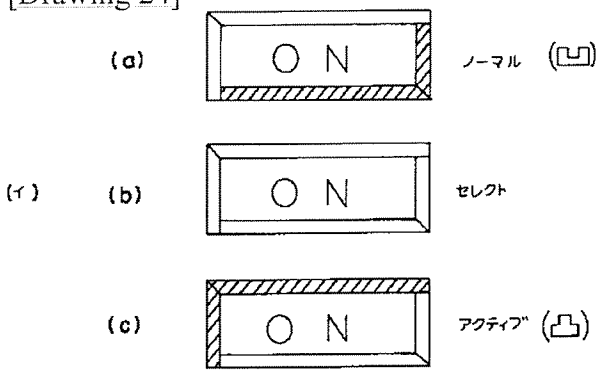
[Drawing 33]

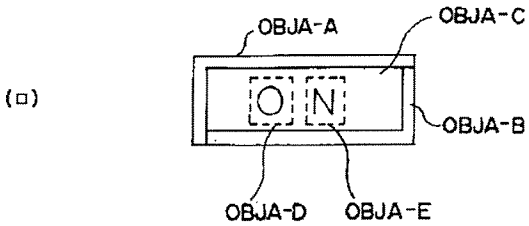


[Drawing 23]

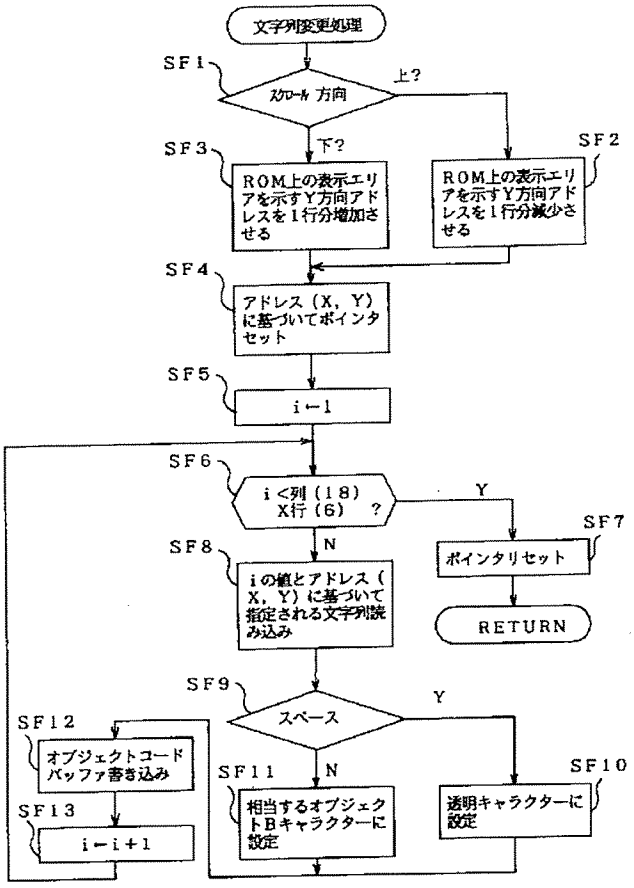


[Drawing 24]

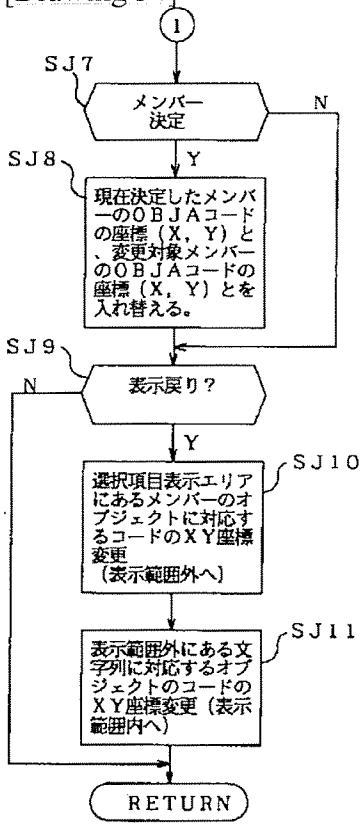




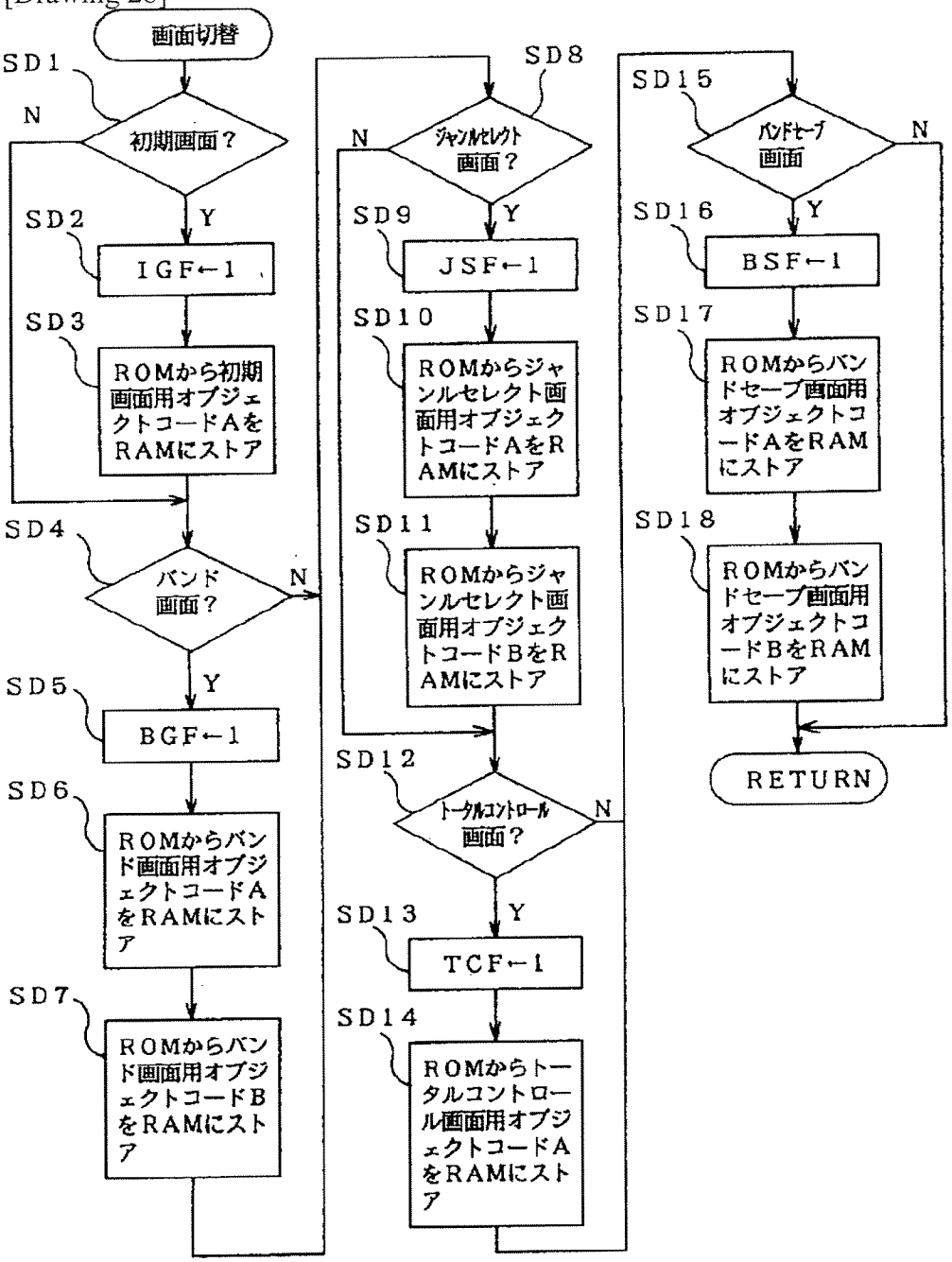
[Drawing 30]


文字列変更処理
SF1
スクロール 方向
上?
下?
SF2
SF3
ROM上の表示エリアを示すY方向アドレスを1行分増加させる
ROM上の表示エリアを示すY方向アドレスを1行分減少させる
SF4
アドレス（X，Y）に基づいてポインタセット
SF5
i←1
SF6
i<列（18）
X行（6） ?
Y
N
SF7
ポインタリセット
RETURN
SF8
iの値とアドレス（X，Y）に基づいて指定される文字列読み込み
SF9
スペース
Y
N
SF10
透明キャラクターに設定
SF11
相当するオブジェクトBキャラクターに設定
SF12
オブジェクトコードバッファ書き込み
SF13
i←i+1

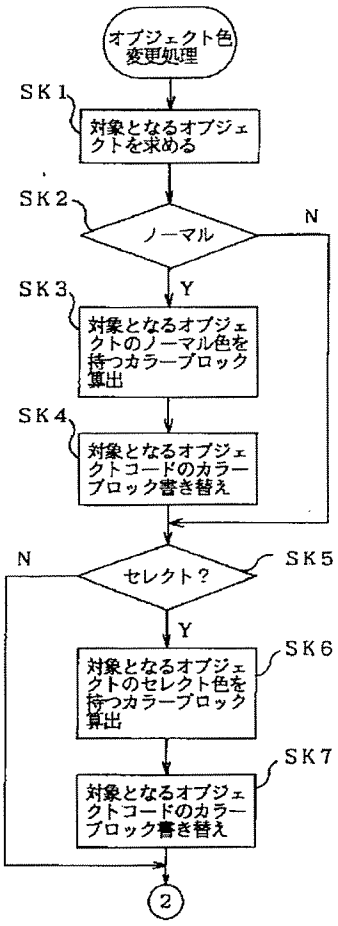
[Drawing 34]


1
SJ7
メンバー決定
N
Y
SJ8
現在決定したメンバーのOBJAコードの座標（X，Y）と、変更対象メンバーのOBJAコードの座標（X，Y）とを入れ替える。
SJ9
N
表示戻り?
Y
SJ10
選択項目表示エリアにあるメンバーのオブジェクトに対応するコードのXY座標変更（表示範囲外へ）
SJ11
表示範囲外にある文字列に対応するオブジェクトのコードのXY座標変更（表示範囲内へ）
RETURN

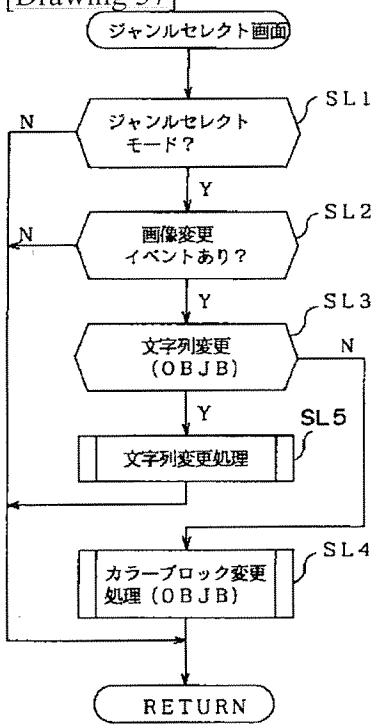
[Drawing 28]



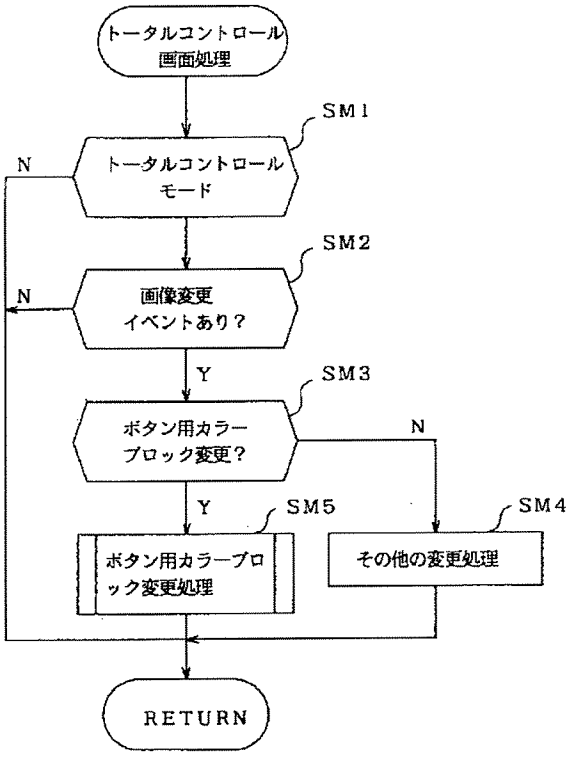
[Drawing 35]



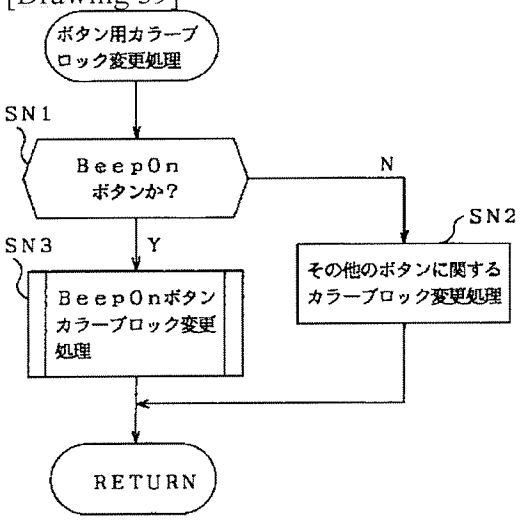
[Drawing 37]



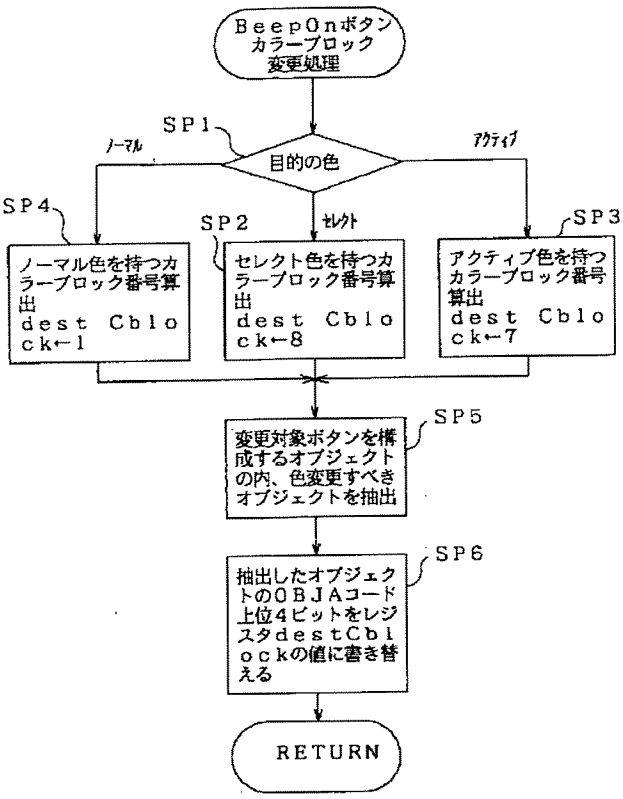
[Drawing 38]



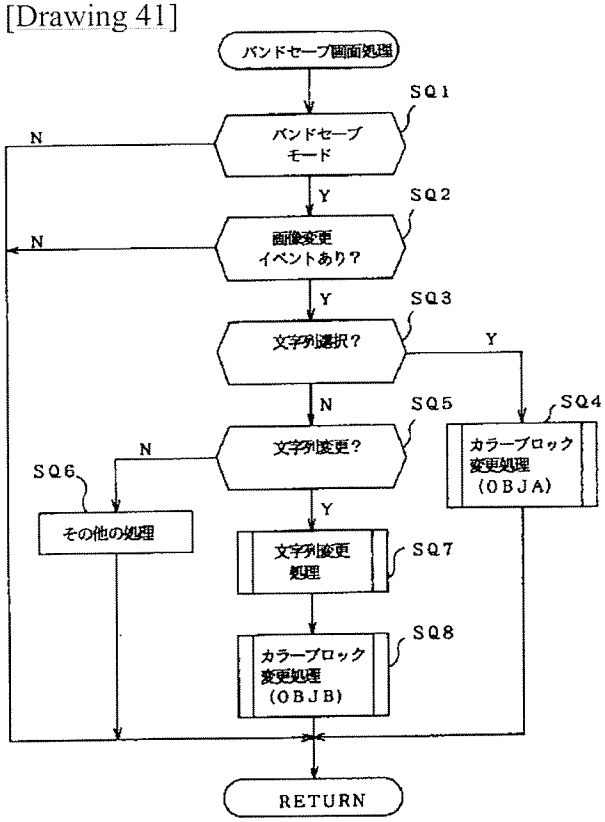
[Drawing 39]



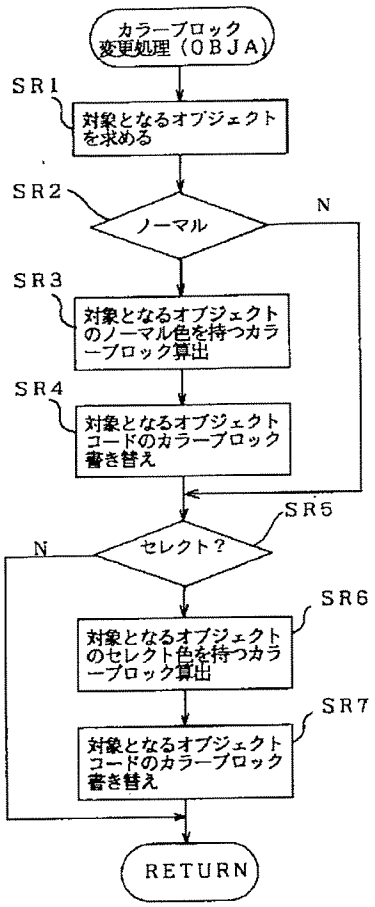
[Drawing 40]



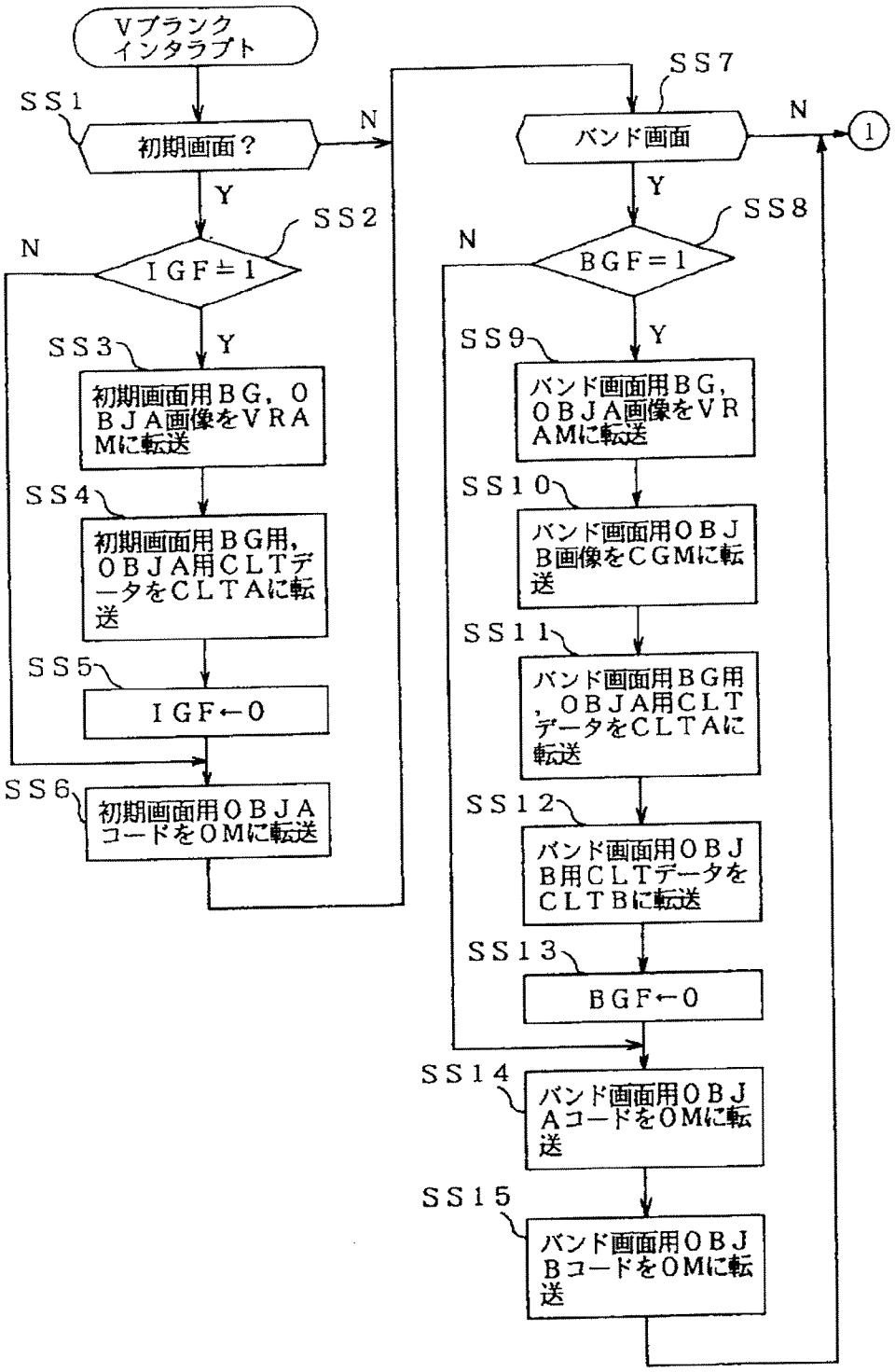
[Drawing 41]



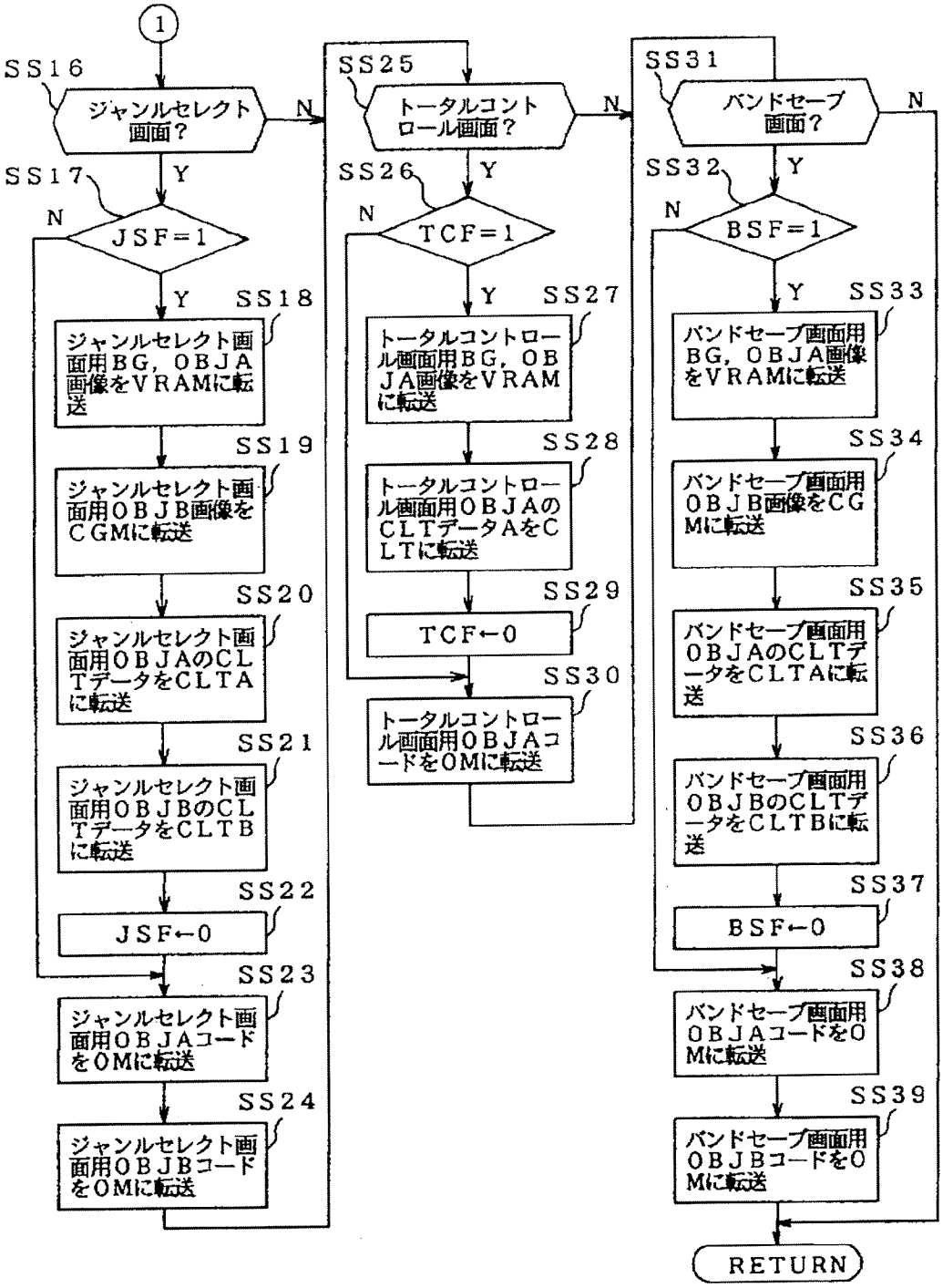
[Drawing 42]


カラーブロック
変更処理（ＯＢＪＡ）
SR1
対象となるオブジェクト
を求める
SR2
ノーマル
N
SR3
対象となるオブジェクト
のノーマル色を持つカラ
ーブロック算出
SR4
対象となるオブジェクト
コードのカラーブロック
書き替え
SR5
セレクト？
N
SR6
対象となるオブジェクト
のセレクト色を持つカラ
ーブロック算出
SR7
対象となるオブジェクト
コードのカラーブロック
書き替え
RETURN

[Drawing 43]


Vブランク
インタラプト
SS1
初期画面？
N
Y
SS2
IGF＝1
N
Y
SS3
初期画面用ＢＧ，ＯＢＪＡ画像をＶＲＡＭに転送
SS4
初期画面用ＢＧ用，ＯＢＪＡ用ＣＬＴデータをＣＬＴＡに転送
SS5
IGF←0
SS6
初期画面用ＯＢＪＡコードをＯＭに転送
SS7
バンド画面
N
1
Y
SS8
BGF＝1
N
Y
SS9
バンド画面用ＢＧ，ＯＢＪＡ画像をＶＲＡＭに転送
SS10
バンド画面用ＯＢＪＢ画像をＣＧＭに転送
SS11
バンド画面用ＢＧ用，ＯＢＪＡ用ＣＬＴデータをＣＬＴＡに転送
SS12
バンド画面用ＯＢＪＢ用ＣＬＴデータをＣＬＴＢに転送
SS13
BGF←0
SS14
バンド画面用ＯＢＪＡコードをＯＭに転送
SS15
バンド画面用ＯＢＪＢコードをＯＭに転送

[Drawing 44]



[Translation done.]